

ePOS-Device XML User's Manual

Overview

Describes the features and development environment.

Building Environment

Describes environment building for ePOS-Device Service I/F.

Programming Guide

Describes how to write programs in Web application development.

ePOS-Device XML

Describes the ePOS-Device XML.

Device Control Script

Describes the device control script.

Sample Program

Describes the sample program.

Appendix

Cautions

- No part of this document may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior written permission of Seiko Epson Corporation.
- The contents of this document are subject to change without notice. Please contact us for the latest information.
- While every precaution has taken in the preparation of this document, Seiko Epson Corporation assumes no responsibility for errors or omissions.
- Neither is any liability assumed for damages resulting from the use of the information contained herein.
- Neither Seiko Epson Corporation nor its affiliates shall be liable to the purchaser of this product or third parties for damages, losses, costs, or expenses incurred by the purchaser or third parties as a result of: accident, misuse, or abuse of this product or unauthorized modifications, repairs, or alterations to this product, or (excluding the U.S.) failure to strictly comply with Seiko Epson Corporation's operating and maintenance instructions.
- Seiko Epson Corporation shall not be liable against any damages or problems arising from the use of any options or any consumable products other than those designated as Original Epson Products or Epson Approved Products by Seiko Epson Corporation.

Trademarks

EPSON is a registered trademark of Seiko Epson Corporation.

Exceed Your Vision and ESC/POS are registered trademarks or trademarks of Seiko Epson Corporation.

Windows® and Internet Explorer® are either registered trademarks or trademarks of Microsoft Corporation in the United States and other countries.

Android™ is trademark of Google Inc. in the United States and other countries.

Apple®, Mac OS® and Xcode® iPhone®, iPad® are either registered trademarks or trademarks of Apple Inc. in the United States and other countries.

IOS® is registered trademarks or trademarks of Cisco in the United States and other countries.

Java™ is a registered trademark of Oracle Corporation, its subsidiaries, and affiliates in the U.S. and other countries.

Eclipse® is a trademark or registered trademark of Eclipse Foundation, Inc.

QR Code® is a registered trademark of DENSO Wave Incorporated.

All other trademarks are the property of their respective owners and used for identification purpose only.

ESC/POS® Proprietary Command System

Epson took the initiative by introducing ESC/POS, a proprietary POS printer command system, which includes patented or patent pending commands and enables versatile POS system construction with high scalability.

Compatible with all types of Epson POS printers and displays, this proprietary control system also offers the flexibility to easily make future upgrades. Its popularity is worldwide.

ESC/POS is designed to reduce the processing load on the host computer in POS environments. It comprises a set of highly functional and efficient commands that enables the full realization of the potential of printers.

© Seiko Epson Corporation 2012-2015. All rights reserved.

For Safety

Key to Symbols

The symbols in this manual are identified by their level of importance, as defined below. Read the following carefully before handling the product.

	Provides information that must be observed to avoid damage to your equipment or a malfunction.
	Provides important information and useful tips.

Restriction of Use

When this product is used for applications requiring high reliability/safety such as transportation devices related to aviation, rail, marine, automotive etc.; disaster prevention devices; various safety devices etc; or functional/precision devices etc, you should use this product only after giving consideration to including fail-safes and redundancies into your design to maintain safety and total system reliability. Because this product was not intended for use in applications requiring extremely high reliability/safety such as aerospace equipment, main communication equipment, nuclear power control equipment, or medical equipment related to direct medical care etc, please make your own judgment on this product's suitability after a full evaluation.

About this Manual

Aim of the Manual

This manual is aimed to provide developers with information required for developing and designing applications using ePOS-Device XML.

The TM-Intelligent series in this manual is a generic term for the TM-DT series and TM-i series.

The TM-DT series is a generic term for the following printers.

- TM-H6000IV-DT
- TM-T70II-DT
- TM-T88V-DT

The TM-i series is a generic term for the following printers.

- TM-T20II-i
- TM-T82II-i
- TM-T83II-i
- TM-T70-i (TM-i firmware Ver.4.0 or later)
- TM-T88V-i (TM-i firmware Ver.4.0 or later)
- TM-U220-i

Manual Content

The manual is made up of the following sections:

Chapter 1	Overview
Chapter 2	Building Environment
Chapter 3	Programming Guide
Chapter 4	ePOS-Device XML
Chapter 5	Device Control Script
Chapter 6	Sample Program
Appendix	List of KeyCode Printer specifications

Contents

■ For Safety	3
Key to Symbols	3
■ Restriction of Use	3
■ About this Manual	4
Aim of the Manual.....	4
Manual Content	4
■ Contents	5

Overview 11

■ ePOS-Device XML	11
Features	12
System Configuration Example	13
How to Use the Communication Box	14
Specifying the Print Job ID from the Application.....	17
Spooler Function	18
■ Operating Environment	21
Application Environment	21
Application Terminal	21
TM-Intelligent	21
ePOS-Device Service Versions.....	21
TM Printer	22
Customer Display.....	23
Peripheral Device	23
■ Contents in the Package	24
Manual.....	24
Sample Program	24
Download.....	24
■ Restrictions	25

Building Environment 27

■ Workflow	27
TM-DT Series	27
TM-i Series.....	28
■ Registration of Device	29
TM-DT Series	29
TM-i Series.....	31

Programming Guide 33

■ Programming Flow	33
If Reconnecting to the TM-Intelligent when the Network was Cut Off	34
Programming flow for the Communication Box	35
■ Using the Spooler Function	38
■ Service Interface Specifications	39
Format	39
■ Communication Data Example	40
Communication Box Message Example	43
■ Array of <data> elements in <device_data> message	46

ePOS-Device XML 47

■ ePOS-Device XML List	47
Message	47
Message Data for Communication Box	48
Message Data for Each Device	48
■ Messages	51
<connect>	51
<reconnect>	52
<disconnect>	53
<admin_info>	54
<open_commbox>	56
<close_commbox>	58
<commbox_data>	59
<open_device>	60
<close_device>	62
<device_data>	64
<error>	65
■ Message Data for Communication Box	66
<type>getcommhistory</type>	66
<type>send</type>	68
<type>onreceive</type>	69
■ Message Data for Device Hub Terminal	70
<type>shutdown</type>	70
<type>onshutdown</type>	70
<type>restart</type>	71
<type>onrestart</type>	71
■ Message Data for the Barcode Scanner	72
<type>ondata</type>	72
■ Message Data for Printers	73
<type>print</type>	73
<type>onxmlresult</type>	74
Print Job ID	75

■ Message Data for Hybrid Printers	77
<type>lock</type>.....	77
<type>unlock</type>.....	77
<type>print</type>.....	78
<type>onxmlresult</type>.....	78
<type>slippprint</type>.....	79
<type>slipcanceled</type>.....	79
<type>endorseprint</type>.....	80
<type>endorsecanceled</type>.....	80
<type>micread</type>.....	81
<type>micrcleaning</type>.....	82
<type>micrcanceled</type>.....	82
<type>eject</type>.....	82
<type>onreceive</type>.....	83
■ Printer Control XML	85
<epos-print>.....	85
<response>.....	87
<text>.....	90
<feed>.....	96
<image>.....	98
<logo>.....	100
<barcode>.....	101
<symbol>.....	105
<hline>.....	110
<vline-begin>.....	111
<vline-end>.....	112
<page>.....	113
<area>.....	114
<direction>.....	115
<position>.....	116
<line>.....	117
<rectangle>.....	118
<cut>.....	119
<pulse>.....	120
<sound>.....	121
<command>.....	123
<layout>.....	124
<recovery>.....	127
<reset>.....	127
■ Message Data for the Customer Display	128
<type>display</type>.....	128
<type>onxmlresult</type>.....	129
■ Customer Display Control XML	130
<epos-display>.....	130
<response>.....	131
<window>.....	132
<text>.....	134
<cursor>.....	135
<blink>.....	136
<brightness>.....	137
<marquee>.....	138
<clock>.....	139
<clear>.....	139
<reset>.....	139
<command>.....	139

■ Message Data for the POS Keyboard.....	140
<type>onkeypress</type>	140
<type>setprefix</type>	141
<type>onstring</type>	142
<type>setMSRPrefix</type>	143
<type>ondata</type>	144
■ Message Data for the Serial Device.....	145
<type>sendcommand</type>.....	145
<type>oncommandreply</type>	146

Device Control Script 147

■ Programming.....	147
Using Device Control Script.....	147
Configuration of Device Control Script.....	148
■ List of Device Control Script API.....	150
ClientConnection object	150
DeviceConnection object	150
Device Control Script Name object.....	150
■ ClientConnection Object.....	151
send	151
■ DeviceConnection Object.....	152
send	152
■ Device Control Script Name Object	153
onDeviceData event (Input Device That Can Run with a HID Driver)	153
onDeviceData event (Serial Communication Device)	153
Any event	154

Sample Program 155

■ Outline.....	155
■ Building Environment for Android	156
Environment	156
Flow of environment construction for TM-DT Series	157
Flow of environment construction for TM-i Series	159
Installing Java SE Development Kit.....	161
Setting Up Eclipse	163
Setting Up Android SDK Manager	165
Installing ADT Plugin.....	170
Importing the Sample Program	174
Setting the Barcode Scanner.....	176
■ Building Environment for iOS	177
Environment	177
Flow of environment construction for TM-DT Series	178
Flow of environment construction for TM-i Series	179
Setting the Barcode Scanner.....	180
Setting the Sample Program	181



■ Executing Sample Program.....	182
Android.....	182
■ How to Use the Sample Program	187

Appendix.....	189
----------------------	------------

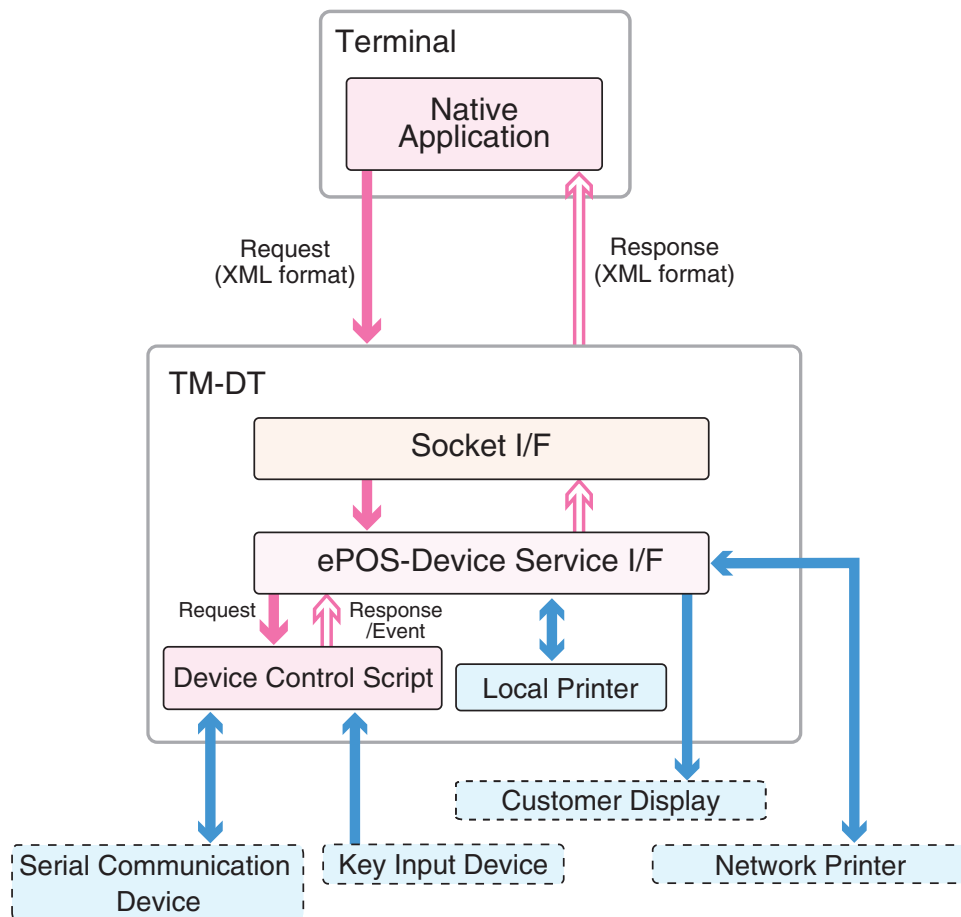
■ List of KeyCode.....	189
■ Printer specifications.....	191
TM-H6000IV-DT/ TM-H6000IV	191
TM-L90	193
TM-P20.....	195
TM-P60II/ TM-P60II with Peeler.....	197
TM-P80.....	199
TM-T20	200
TM-T20II-i/ TM-T20II	201
TM-T70-i/ TM-T70	203
TM-T70II-DT/ TM-T70II	204
TM-T82II-i/ TM-T82II	205
TM-T83II-i/ TM-T83II	207
TM-T88IV	209
TM-T88V-DT/ TM-T88V-i/ TM-T88V.....	211
TM-T90	213
TM-U220-i/ TM-U220.....	215



Overview

This chapter describes the features of and the specifications for ePOS-Device XML.

ePOS-Device XML



ePOS-Device XML is a command system that defines the function to control various types of POS peripheral devices (including this product's printer) connected to TM-Intelligent using XML. Application in devices such as personal computers, smartphones, and tablet computers creates a request message in XML format and send it to TM-Intelligent using socket communication. ePOS-Device Service installed in TM-Intelligent executes control over peripheral devices by interrupting a request message and returns a response.

Features

- ❑ No need to prepare any device (such as a PC) to act as a controller to control the peripheral devices.
- ❑ Devices that run with the OS-standard driver can be used with a device control script without any drivers installed.
- ❑ Accessing a device automatically locks the device exclusively.
Even if accessed from multiple terminals simultaneously, the device is not under multiple controls. When the terminal that has controlled the device releases the device, the device becomes controllable from another terminal.
- ❑ HTTP and HTTPS ^{*2} communications are supported.
- ❑ If the network is cut off, you can easily re-establish the connection.
- ❑ Data can be sent and received between web applications.
For details, refer to [How to Use the Communication Box\(p.14\)](#).
- ❑ Printing requests can be made to the printer by specifying the printing job ID. ^{*3}
For details, refer to [Specifying the Print Job ID from the Application\(p.17\)](#).
- ❑ Printing data can be sent to the printer without waiting for a printing complete response from the printer. (Spooler) ^{*2}
For details, refer to [Spooler Function\(p.18\)](#).
- ❑ Print data can be sent to another printer if the printer sends an error response (PAPER END, COVER OPEN, etc.) or does not return a response. (Print forwarding) ^{*3}
For details, refer to [Forwarding\(p.20\)](#).
- ❑ You can shut down the TM-DT from the application. ^{*1}
(DeviceHubTerminal object)

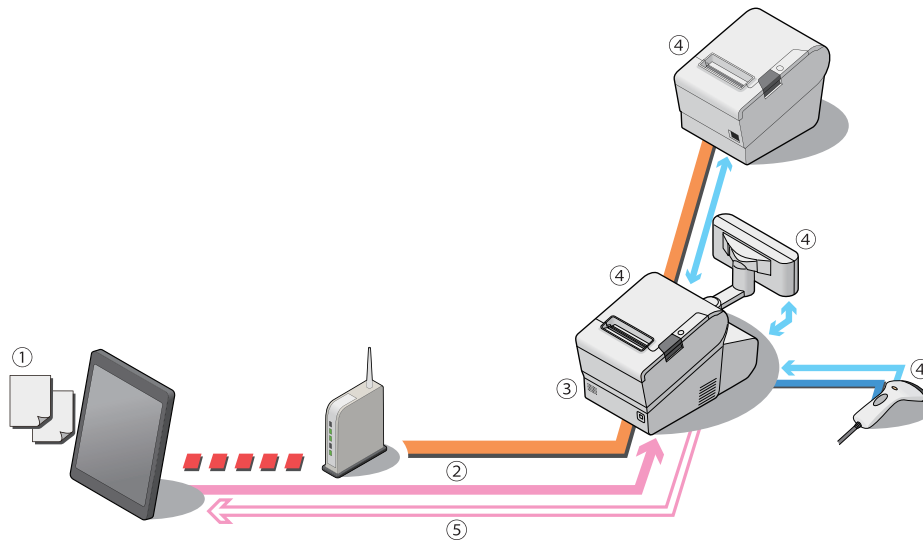
^{*1} TM-DT series is supported only.

^{*2} TM-DT software Ver.2.5 or later, TM-i firmware Ver.4.3 or later

^{*3} ePOS-Device Service Ver.2.6 or later

System Configuration Example

System with Application Installed in Tablet Computer



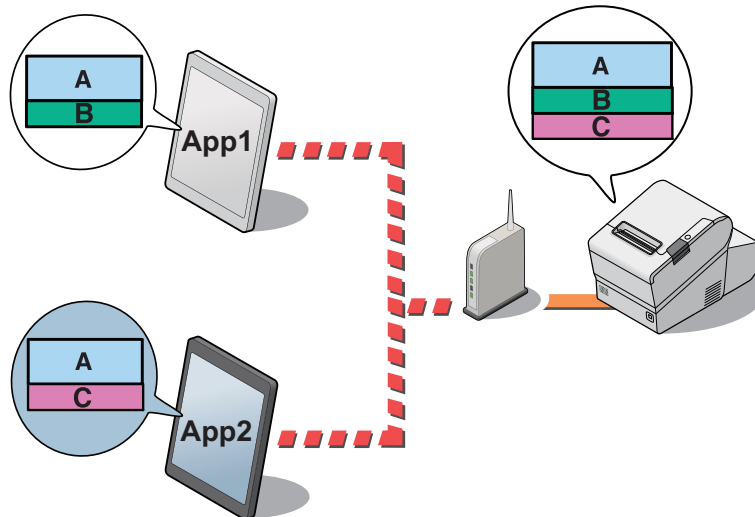
- 1** Place and display the application in the tablet computer.
- 2** The application sends a request message to TM-Intelligent.
- 3** TM-Intelligent sends data to devices that can be controlled.
- 4** The devices and network printers connected to TM-Intelligent are controlled.
- 5** TM-Intelligent returns a response to the application.

How to Use the Communication Box

The Communication Box is a virtual data space that is provided by the ePOS-Device Service I/F for sending and receiving data between applications. It is provided in ePOS-Device Service Ver. 2.5 and later.



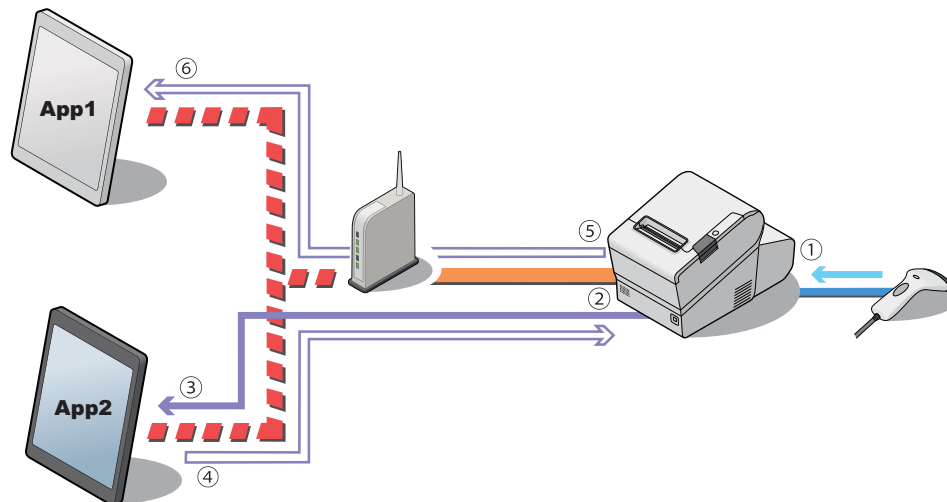
The ePOS-Device Service is providing a virtual space and structure for sending and receiving data between applications. Create the data to be sent and received according to your application's specifications.



The Communication Box can be used in the following ways.

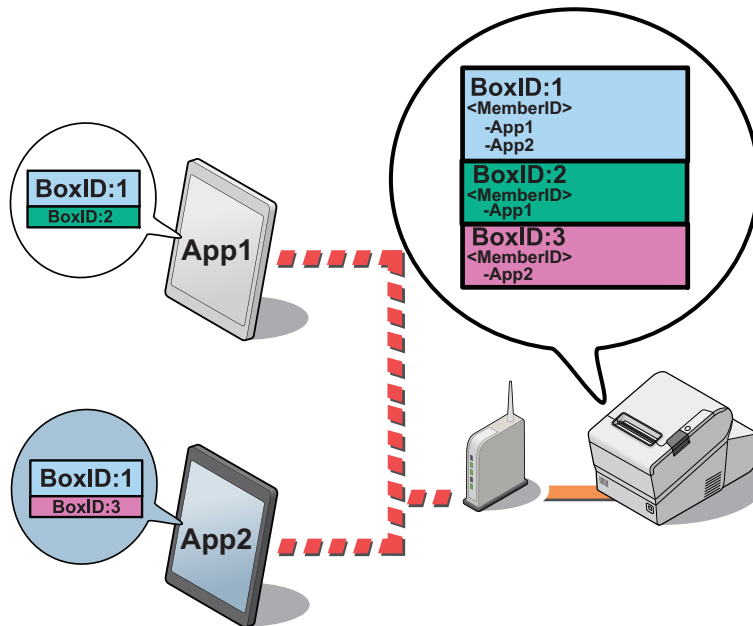
- Using a tablet terminal as a POS display terminal
- Displaying digital signage
- Using a tablet terminal as an entry terminal

Example of data processing using the Communication Box



- 1** The TM-Intelligent receives the barcode data scanned by the scanner.
- 2** The ePOS-Device Service I/F notifies App2 of the barcode data.
- 3** App2 acquires the barcode data and converts it to POS data.
- 4** App2 sends the display data to the Communication Box of the ePOS-Device Service I/F.
- 5** The ePOS-Device Service I/F notifies App1 that the display data is stored in the Communication Box.
- 6** App1 acquires the display data stored in the Communication Box.

Structure of the Communication Box



The ePOS-Device Service I/F manages Communication Boxes with Box IDs (BoxID in the figure above).

Data can be sent and received between applications belonging to the Communication Box.

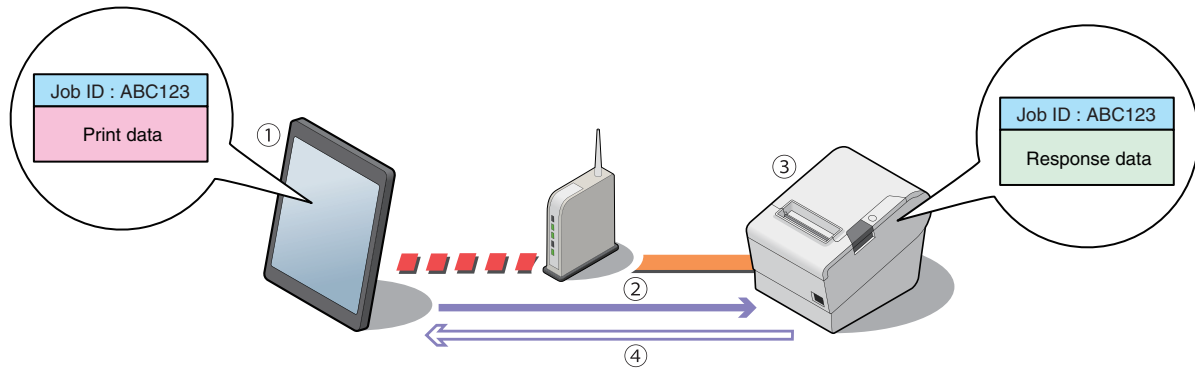
In the figure above, applications App1 and App2 can send and receive data with one another using the BoxID:1 Communication Box.

Specifications of the Communication Box

Maximum number of Communication Boxes that can be created	20
Maximum number of applications that can belong to one Communication Box	20
Space of transfer history that one Communication Box can maintain	10240 Byte
Size of data that can be sent at one time	1024 Byte

Specifying the Print Job ID from the Application

A response containing the specified print job ID will be returned when sending a request from the application by specifying the print job ID. (ePOS-Device Service Ver. 2.6 or later versions)



Spooler Function

The spooler function is a function used to temporarily store printing data received from the application in the TM-Intelligent to carry out background printings sequentially. (ePOS-Device Service Ver. 2.6 or later versions)

TM-Intelligent return response data to the application the moment the printing data enters the spooler to make it possible for applications to proceed to the next process without waiting for a printing complete response.

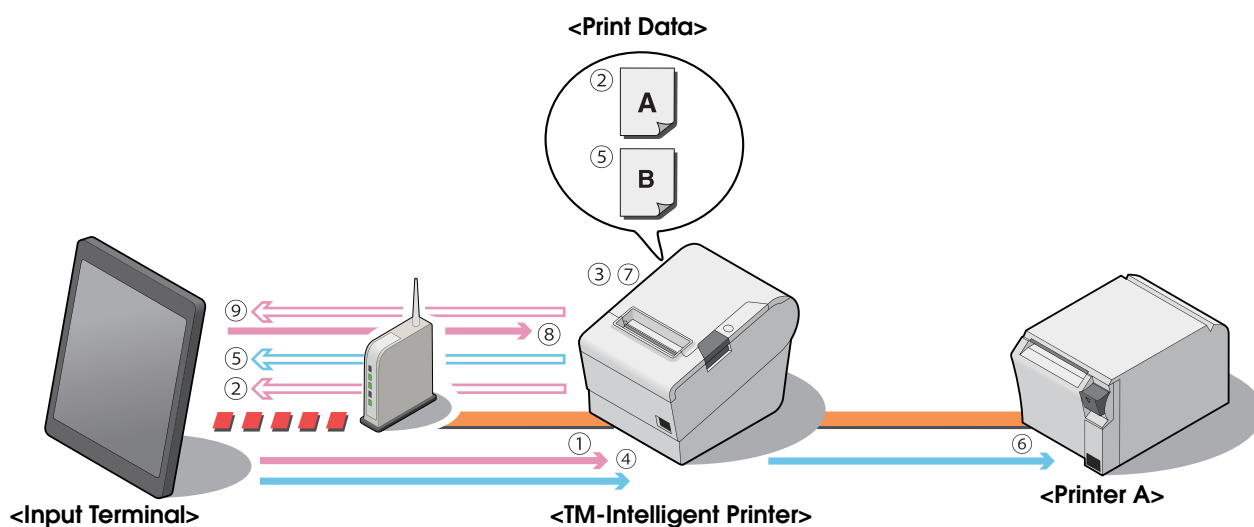
The application can use the job ID included in the response data to send an inquiry on the printing status. EPSON TMNet WebConfig is used to configure the spooler.

Perform the following settings. For details, refer to the Technical Reference Guide for each TM-Intelligent printer.

- Settings to enable the spooler
- Print process retry settings

Forwarding can be used to send a printing job to a different printer if the printer does not respond to retry processing. For details, refer to [Forwarding\(p.20\)](#).

Example



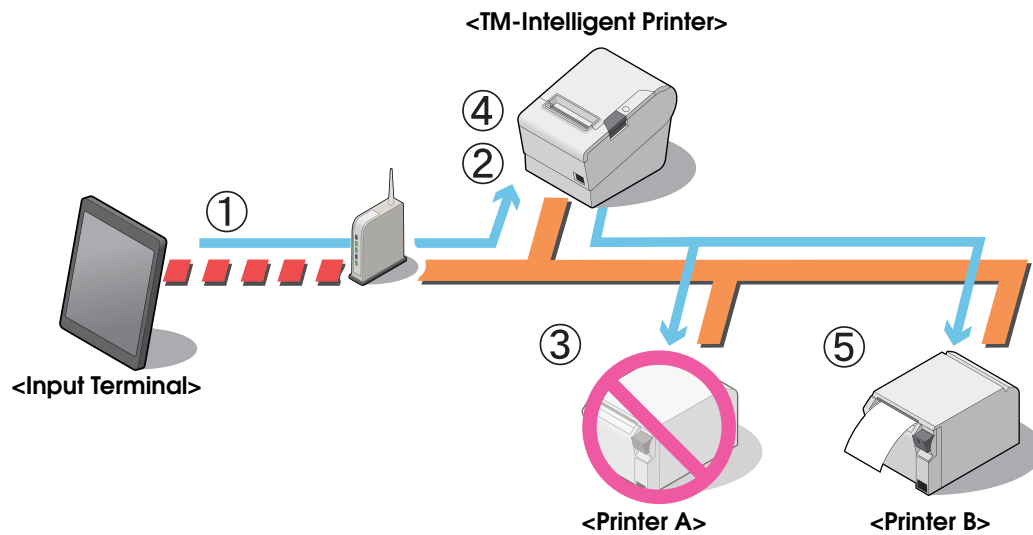
- 1 Send print data A to a TM-Intelligent (ePOS-Device Service I/F) from the input terminal.
- 2 The TM-Intelligent will save the print data A to the spooler and will return response data to the input terminal.
- 3 The TM-Intelligent will retrieve the print data and issue a print command to another TM-Intelligent.

- 4** Print data B will be sent to the TM-Intelligent from the input terminal.
- 5** The TM-Intelligent will save the print data B to the spooler and will return response data to the input terminal.
- 6** The TM-Intelligent will retrieve the print data B and issue a print command to printer A.
- 7** Print data A is complete.
- 8** A query on print data A results will be sent to the TM-Intelligent from the input terminal.
- 9** The TM-Intelligent will return response data back to the input terminal signifying that print data A completed without error.

Forwarding



EPSON TMNet WebConfig is used to configure forwarding settings. For details, refer to the Technical Reference Guide for each TM-Intelligent Printer.



- 1** Send print data to ePOS-Device Service I/F from the input terminal.
- 2** The TM-Intelligent issues a print command to printer A.
- 3** Printer A does not respond.
- 4** The TM-Intelligent reissues a print command to printer A.
If printer A still does not respond to the reissued print command, the print command will be issued to printer B.
- 5** Print data sent from the input terminal will be printed using printer B.

Operating Environment

Application Environment

Environment that allows socket communication and handling of XML documents

- ❑ OS Examples: iOS / Android™ / Windows / Linux / Mac OS

Application Terminal

Devices that allow socket communication as connected to a network

(Example: Smartphones, tablet computers, personal computers)

TM-Intelligent

TM-DT Series

- ❑ TM-H6000IV-DT
- ❑ TM-T70II-DT
- ❑ TM-T88V-DT

TM-i Series

- ❑ TM-T20II-i
- ❑ TM-T70-i (TM-i firmware Ver.4.0 or later)
- ❑ TM-T83II-i
- ❑ TM-T82II-i
- ❑ TM-T88V-i (TM-i firmware Ver.4.0 or later)
- ❑ TM-U220-i

ePOS-Device Service Versions

The versions of the ePOS-Device service support the following TM-DT Software version and TM-i firmware version.

ePOS-Device Service	TM-DT Software	TM-i Firmware
1.0	1.0	-
2.0	2.0	-
2.2	2.2	-
2.5	2.5	4.0
2.6	-	4.1, 4.3
3.0	3.0	-



For details about how to check the version of TM-i Firmware and of TM-DT Software, refer to the Technical Reference Guide for each TM-Intelligent Printer.

TM Printer

Besides a TM-Intelligent printer, the TM printers below can be controlled.

TM Printer	TM-DT Series	TM-i Series
TM-H6000IV ^{*1}	✓	-
TM-L90 ^{*1}	✓	✓
TM-P20	✓ ^{*3}	✓ ^{*2}
TM-P60II	✓	✓
TM-P60II Peeler	✓	✓
TM-P80	✓ ^{*3}	✓
TM-T20 ^{*1}	✓	✓
TM-T20II-i	-	✓
TM-T20II ^{*1}	✓ ^{*3}	✓
TM-T70-i (TM-i firmware Ver.4.0 or later)	-	✓
TM-T70 ^{*1}	✓	✓
TM-T70II ^{*1}	✓	✓
TM-T82II-i	-	✓
TM-T82II ^{*1}	✓ ^{*3}	✓
TM-T83II-i	-	✓ ^{*2}
TM-T83II ^{*1}	-	✓ ^{*2}
TM-T88IV ^{*1}	-	✓
TM-T88V-i (TM-i firmware Ver.4.0 or later)	-	✓
TM-T88V ^{*1}	✓	✓
TM-T90 ^{*1}	✓	✓
TM-U220-i	-	✓
TM-U220 ^{*1}	✓	✓

^{*1} Ethernet model (UB-E02/ UB-E03), or wireless LAN model (UB-R03/ UB-R04).

Other interface models are not supported.

^{*2} TM-i firmware Ver.4.1 or later versions supported.

^{*3} TM-DT software Ver.3.0 or later versions supported.

Customer Display

The following customer display is available:

- ❑ TM-T88V-DT
 - DM-D110 for TM-T88V-DT model
 - DM-D110 RS-232/USB model, USB model
- ❑ Other TM-DT series
 - DM-D110 RS-232/USB model, USB model
- ❑ TM-i series
 - DM-D110 RS-232/USB model, USB model

Peripheral Device

Device	TM-DT Series	TM-i Series
Keyboard device	✓	✓
Barcode scanner	✓	✓
Key input device	✓	✓
Serial communication device	✓	✓
USB device that can provide a control level equivalent to serial communication devices *	✓	-

- * Will require separate installation of a serial-USB conversion driver.
Serial-USB conversion drivers with certain specifications may not be usable.



With the TM-DT Series, a peripheral device can be used with a device control script file developed by the user to match the device.
Also, an API for device control script is available. For details, see [Device Control Script\(p.147\)](#).

Contents in the Package

Manual

- ❑ ePOS-Device XML User's Manual (This Document)
- ❑ TM-H6000IV-DT Technical Reference Guide
- ❑ TM-T70II-DT Technical Reference Guide
- ❑ TM-T88V-DT Technical Reference Guide
- ❑ TM-T20II-i Technical Reference Guide
- ❑ TM-T82II-i Technical Reference Guide
- ❑ TM-T83II-i Technical Reference Guide
- ❑ TM-T70-i (TM-i Firmware Ver.4.0 or later) Technical Reference Guide
- ❑ TM-T88V-i (TM-i Firmware Ver.4.0 or later) Technical Reference Guide
- ❑ TM-U220-i Technical Reference Guide

Sample Program

ePOS-Device_Sample_XML_Vx.x.x.zip

This contains the following.

- ❑ Sample program for Android
- ❑ Sample program for iOS
- ❑ Sample for device control script

File	Target Device
Keyboard_Generic.js	Keyboard
Scanner_Generic.js	Barcode scanner
SimpleSerial_Generic.js	Serial communication device



- A sample program for an Android/ iOS environment is provided in the ePOS-Device XML. For how to use the sample program, refer to [Sample Program\(p.155\)](#).
- "Sample for device control script" is for use with TM-DT. It is not used with the TM-i Series.

Download

For customers in North America, go to the following web site:

<http://www.epsonexpert.com/> and follow the on-screen instructions.

For customers in other countries, go to the following web site:

<https://download.epson-biz.com/?service=pos>

Restrictions

- ❑ The ruled line command is available only when the printer supports the "ruled line command".
- ❑ The drawer and the buzzer cannot be used together.
- ❑ The buzzer function cannot be used if the printer is not provided with the buzzer.
- ❑ When a 2D-code scanner is used, multibyte characters such as Japanese cannot be obtained properly.
- ❑ When 2D-code data contains an ASCII control code (0x00 to 0x1F), control codes cannot be obtained.
- ❑ Keycodes that can be obtained from the keyboard are restricted. For the key codes that can be obtained, refer to [List of KeyCode\(p.189\)](#).
- ❑ Device control script files cannot be registered in a TM-i Series printer.
- ❑ Serial communication devices cannot be used with a TM-i Series printer with the specification of "without Serial port".
- ❑ The PHP function and print spooler function cannot be used simultaneously on a TM-i series.

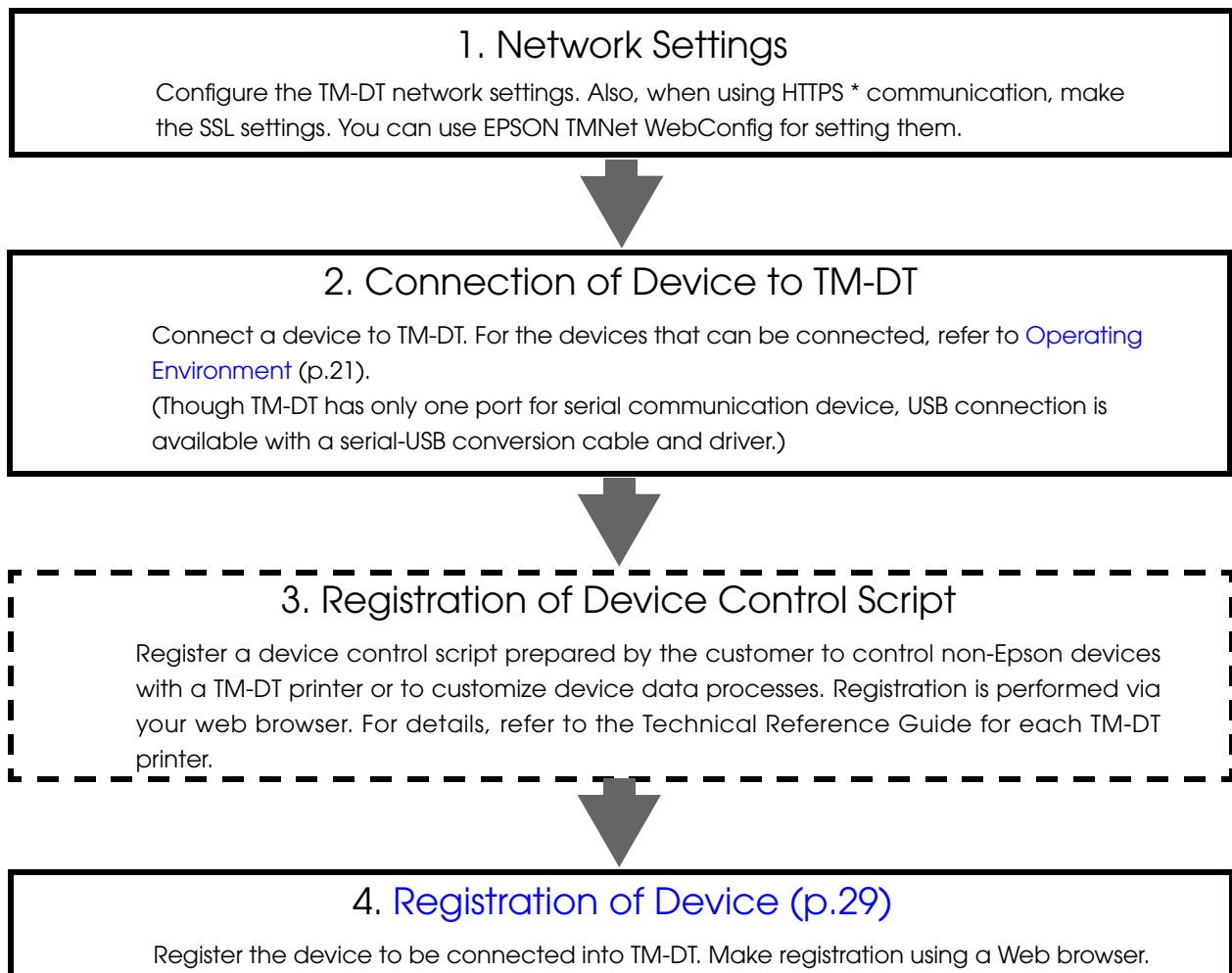


Building Environment

This chapter describes environment building for ePOS-Device Service I/F.

Workflow

TM-DT Series



Necessary step

Optional step

* TM-DT software Ver.2.5 or later, TM-i firmware Ver.4.3 or later

TM-i Series

1.Connection of Device to TM-i

Connect a device to TM-i. For the devices that can be connected, refer to [Operating Environment](#) (p.21).



2. Network Settings

Configure the system and TM-i network settings.

- EPSON TMNet WebConfig
- EpsonNet Config
- Easy Kitting

When using HTTPS * communication, you can use EPSON TMNet WebConfig to make the SSL settings.



3.Registration of Device (p.29)

Register the device to be connected into TM-i. Make registration using a Web browser.

* TM-DT software Ver.2.5 or later, TM-i firmware Ver.4.3 or later

Registration of Device

Register the device to be controlled by ePOS-Device XML into TM-Intelligent. Configure the settings using EPSON TMNet WebConfig.

EPSON TMNet WebConfig can be launched from your web browser.

For details, refer to the Technical Reference Guide for each TM-Intelligent.

TM-DT Series

Configure the following settings for each device connected to the TM-DT printer.

Printer

Setting	Setting Value
Device ID	Enter the ID (any character string) of the printer to be controlled.
Type	Select "Network printer".
Model no.	Select the model of the printer to be controlled.
IP Address	Specify the IP address of the printer for each device ID.
Retry interval	Specify the retry interval after timeout.

Customer Display

Setting	Setting Value
Customer Display	Set whether or not this is used.
Communications settings	Set the communication speed, data bit and parity.
Brightness settings	Set the brightness of the customer display.



The device ID of the customer display is fixed to "local_display".

Key input device

Setting	Setting Value
Device ID	Enter the device ID (any character string).
Device name	Select the device name of the key input device.
Control Script	Select the device control script to be used for the key input device.

Serial communications

Setting	Setting Value
Device ID	Enter the device ID (any character string).
Device name	Select the device name of the serial communication device. The name can be selected from the product names and ports.
Control script	Select the device control script to be used for the serial communication device.
Communication speed(bps)	Set the communication speed of the device.
Data bit	Set the data bit.
Parity	Set the parity.
Stop bit	Set the stop bit.
Flow control	Set the flow control.

Other

Setting	Setting Value
Device ID	Enter the device ID (any character string).
Control script	Select the device control script to be used for the connected device.

TM-i Series

Configure the following settings for each device connected to the TM-i printer.

Printer

Setting	Setting Value
Device ID	Enter the ID (any character string) of the printer to be controlled.
Model	Select the model of the printer to be controlled.
IP Address	Specify the IP address of the printer for each device ID.
Retry interval(ms)	Specify the retry interval after timeout.

Customer Display

Setting	Setting Value
Customer Display	Set whether or not this is used.
Communications settings	Set the communication speed, data bit and parity.
Brightness settings	Set the brightness of the customer display.



The device ID of the customer display is fixed to "local_display".

Key input device

Setting	Setting Value
Device ID	Enter the device ID (any character string).
Device name	Select the device name of the key input device.
Control Script	Select the device control script to be used for the key input device.

Serial communications

Setting	Setting Value
Serial communications	Set whether or not this is used.
Communication speed(bps)	Set the communication speed of the device.
Data bit	Set the data bit.
Parity	Set the parity.
Stop bit	Set the stop bit.
Flow control	Set the flow control.



The device ID of the serial communication device is fixed to "local_serial".

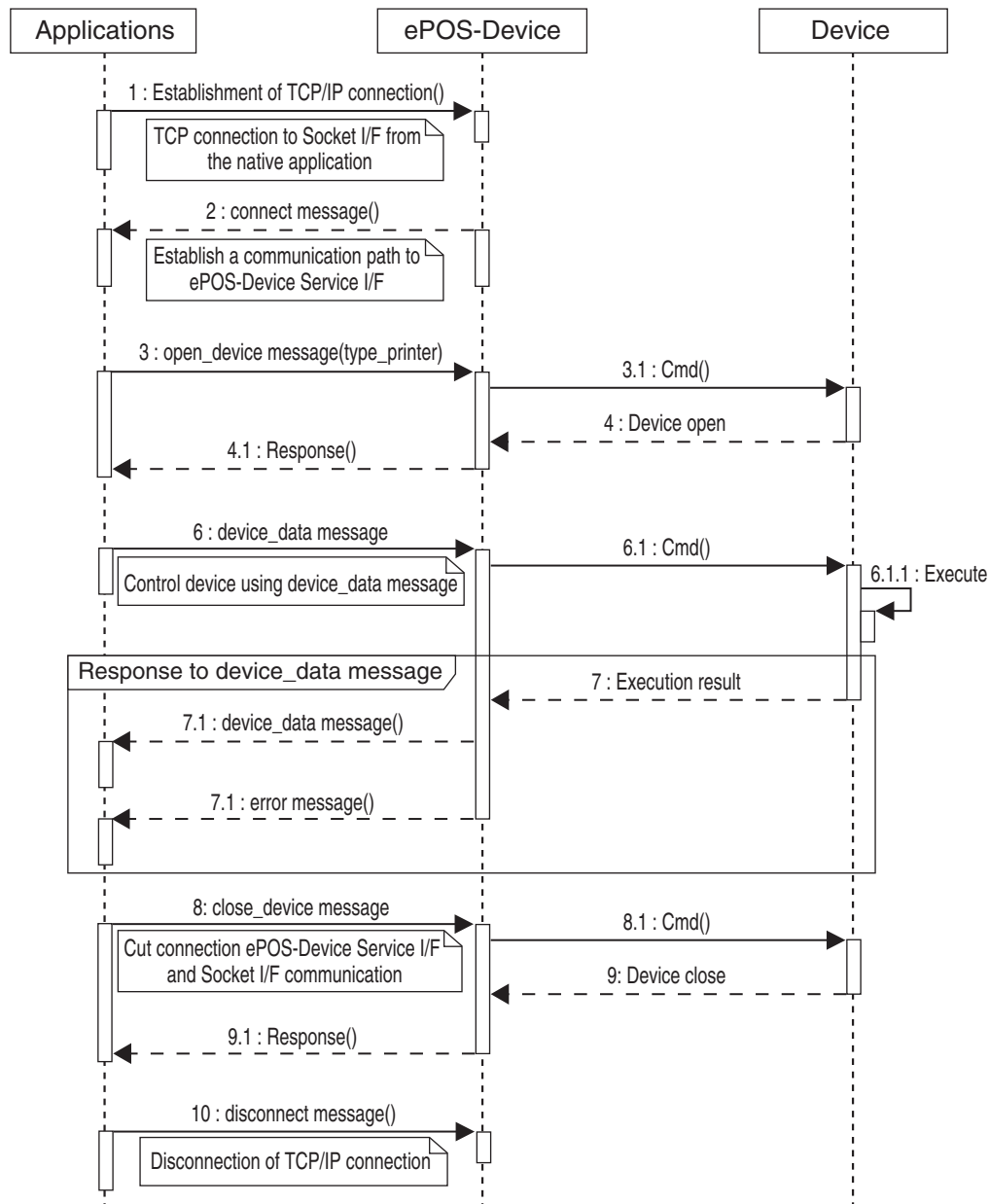


Programming Guide

This chapter describes how to write programs in the application development using ePOS-Device.

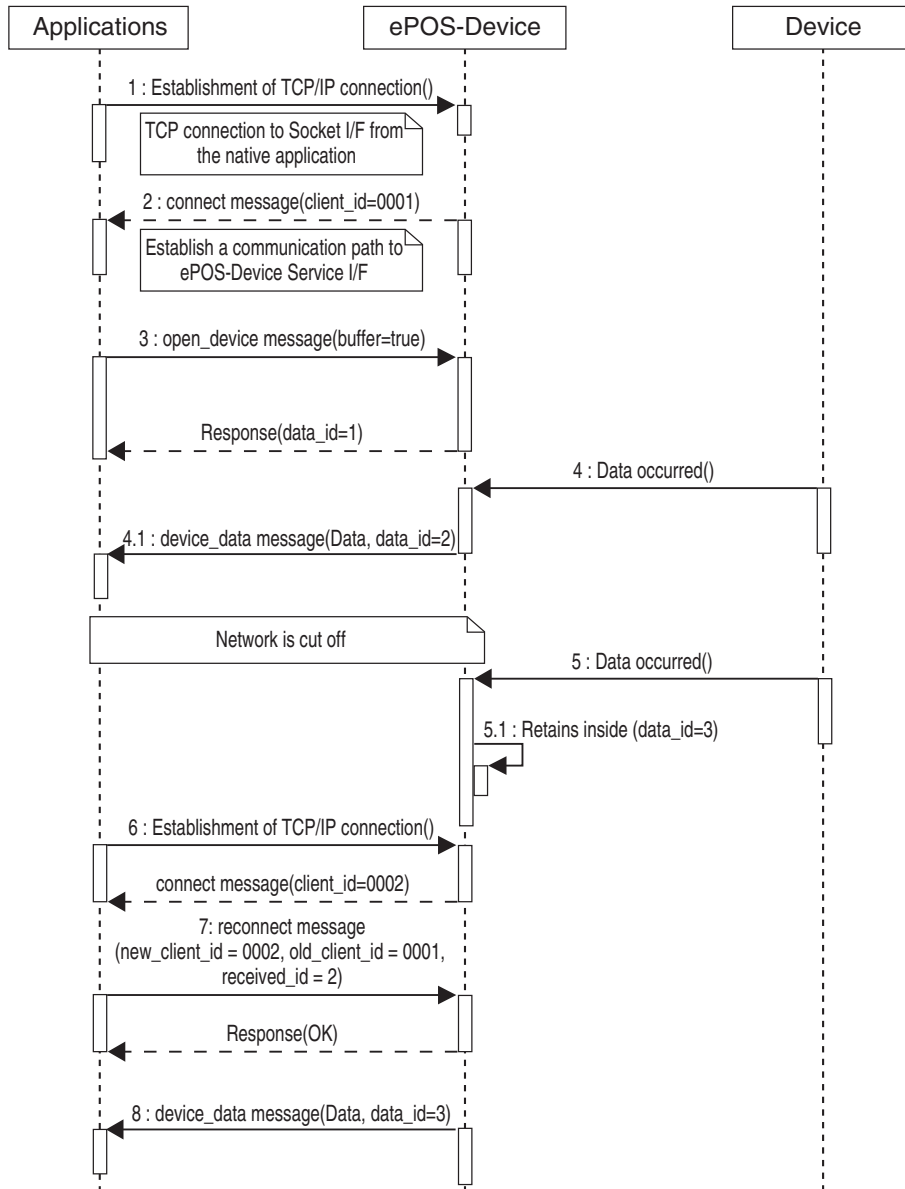
Programming Flow

The basic programming sequence of ePOS-Device XML is as follows:



If Reconnecting to the TM-Intelligent when the Network was Cut Off

If network communication with the TM-Intelligent is cut off, the client application will reconnect. By reconnecting, you can use the devices that were open without reopening them. Also, through the settings for opening, the data to be sent to the client application that occurred while the network was cut off can be received upon reconnection. The following sequence explains the message flow necessary upon reconnection and the necessary parameters.

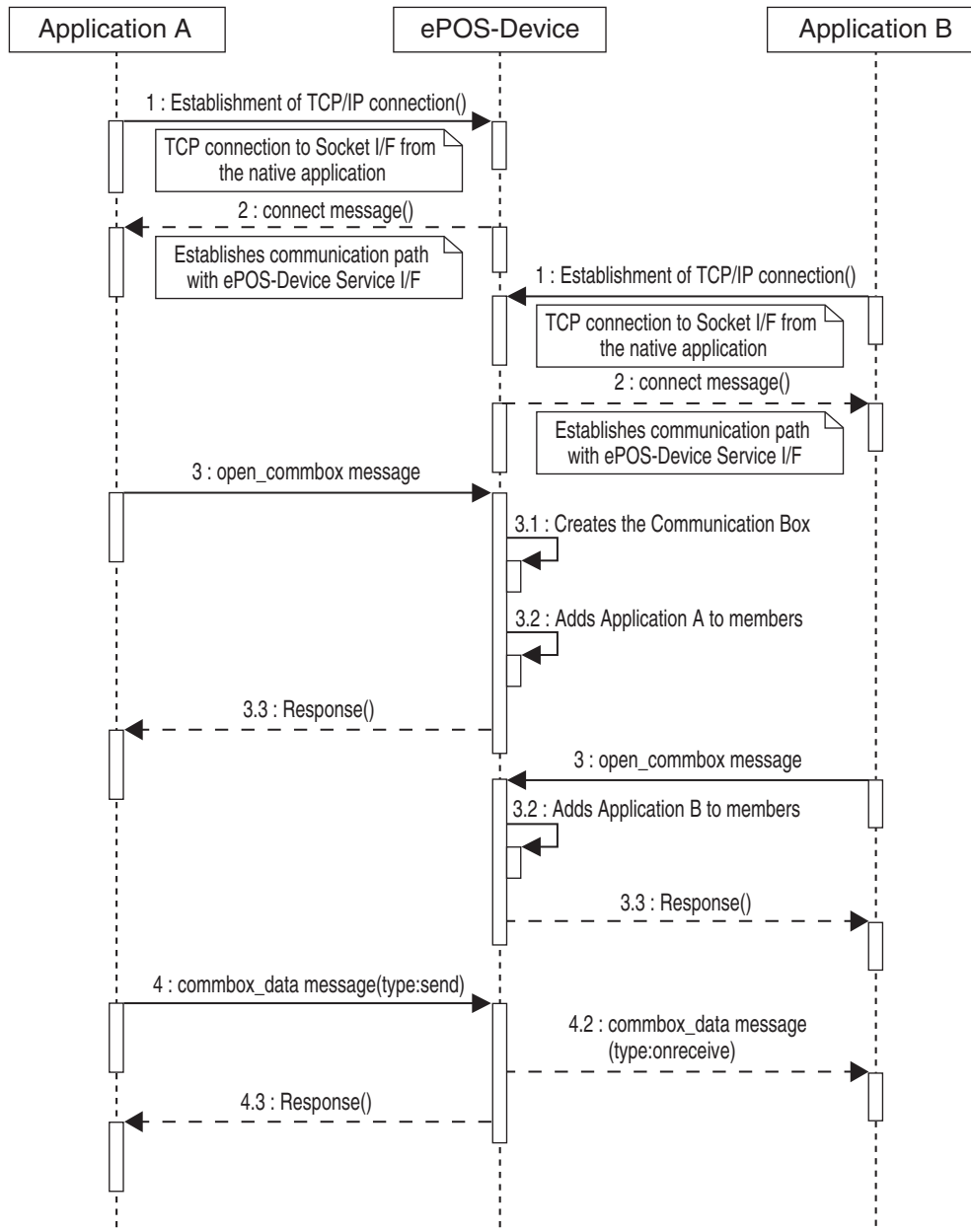


To open an unopen device from the client, be sure to send a <disconnect> message when closing the application. Also, in cases where you cannot send a <disconnect> message when closing the application, store the previous <client_id> in the application's permanent memory beforehand, and upon the next start-up, use the previous <client_id> to send a <disconnect> message.

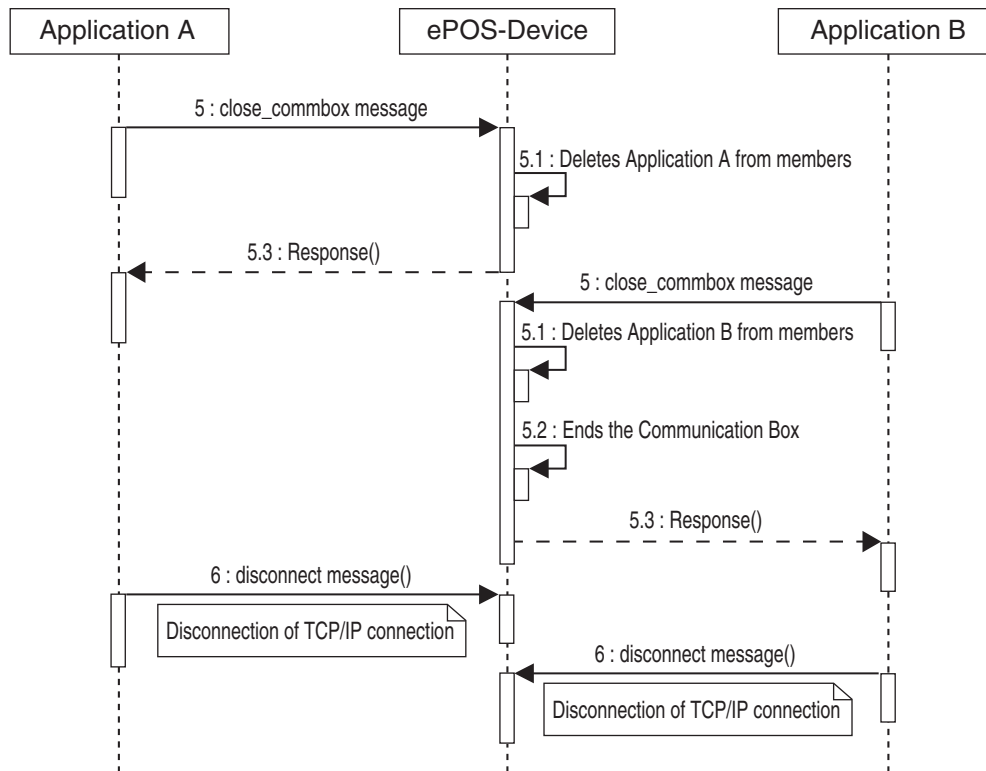
Programming flow for the Communication Box

The following indicates the basic programming sequence using the Communication Box.

(1/2)

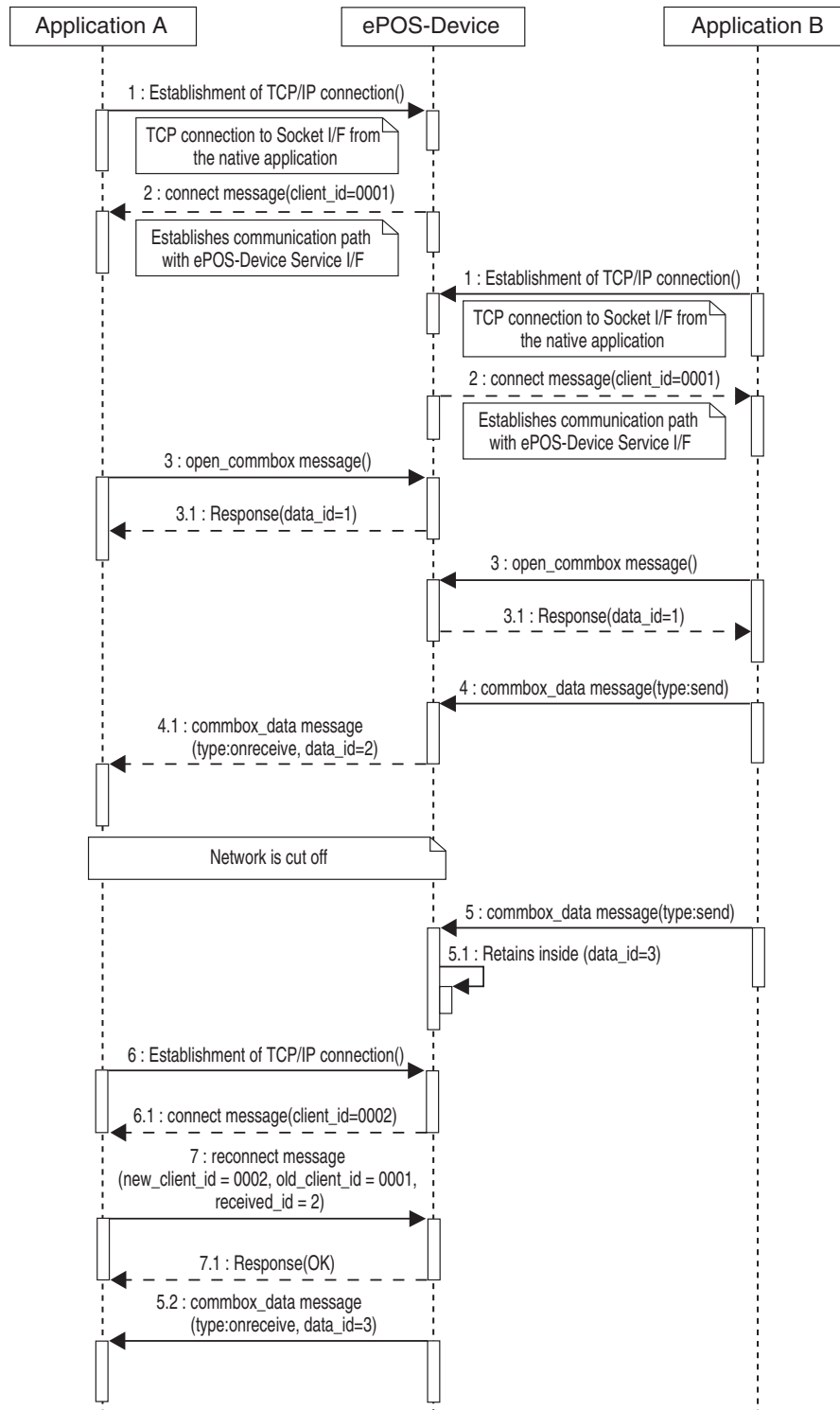


(2/2)



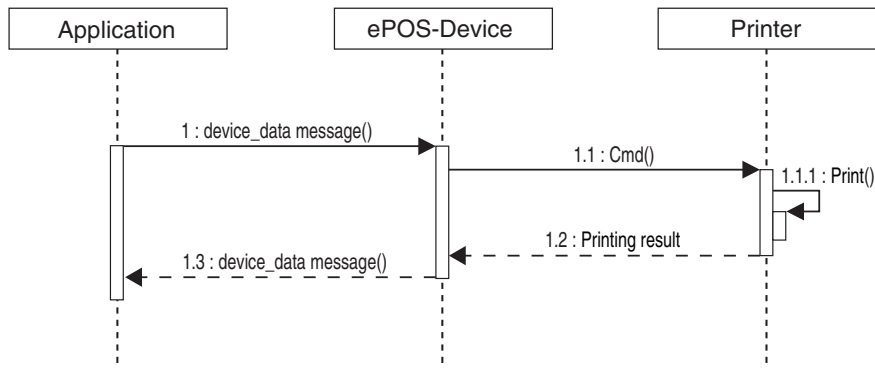
If automatically reconnecting the network

If network communication with the TM-Intelligent was reconnected, the Communication Box sequence is as follows.



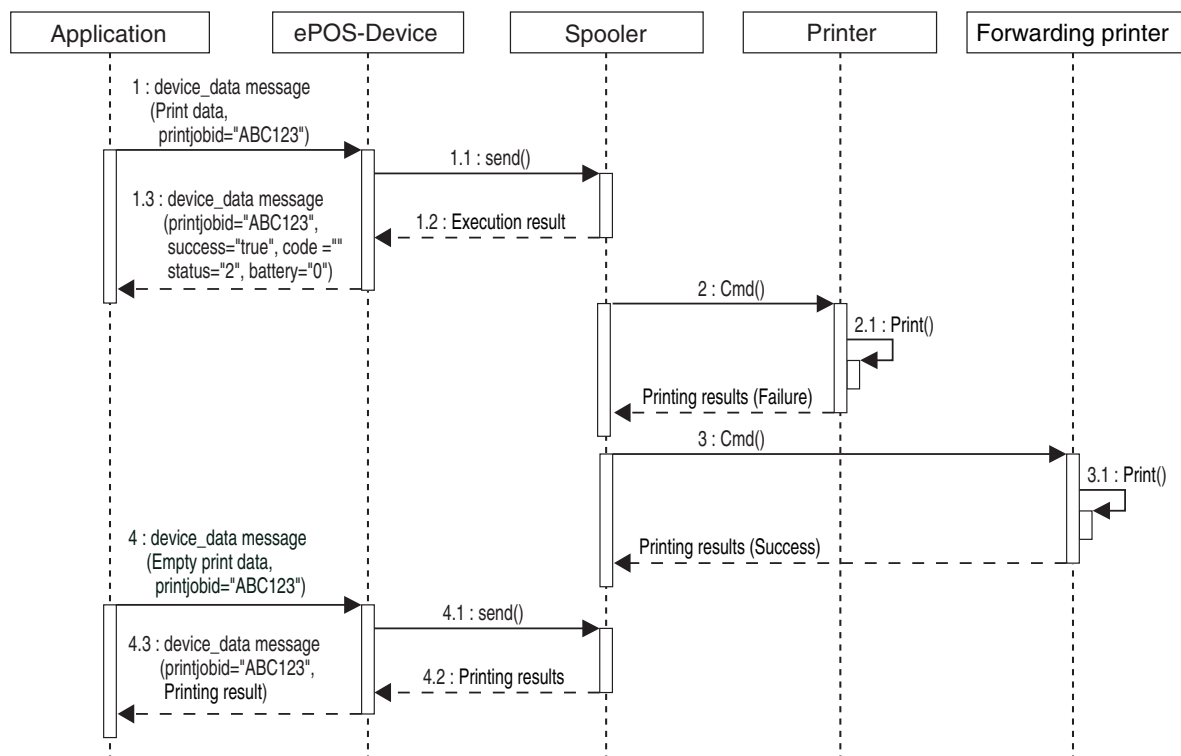
Using the Spooler Function

A printing job will be executed immediately and a response will be returned to the application after printing is complete when sending a printing request to ePOS-Device Service I/F from the application when the spooler function is disabled.



The print data job will be added to the queue and a response will be returned to the application without waiting for printing to complete when sending a printing request to ePOS-Device Service I/F from the application when the spooler function is enabled.

Even if the output printer cannot complete the printing task, ePOS-Device Service I/F does not return an error to the application. Printing on a substitute printer can be done by enabling print forwarding. The application can obtain printing results at a later time by specifying the print job ID and requesting empty print data. Refer to the sequence diagram below.



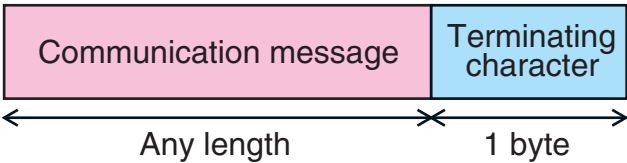
Service Interface Specifications

To enable control using ePOS-Device XML, use the following interface:

Transport layer	Port No.
TCP	8009

Format

The format and description of each data item are as follows:



Data name	Description
Communication message	"UTF-8" character string in XML format Any length
Terminating character	NULL character ("\\0") 1 byte

Communication Data Example

Examples of communication data between the application and ePOS-Device are as shown below, where the NULL character is indicated as "\0".

Applications		ePOS-Device	Description
Establishment of TCP/IP connection	➡		Establishment of connection
	⬅	<pre><connect> <data> <client_id>sock3514555410 </client_id> <protocol_version>2 </protocol_version> </data> </connect>"\0"</pre>	
<admin_info></admin_info>"\0"	➡		Acquisition of the administrator information.
	⬅	<pre><admin_info> <code>OK</code> <data> <admin_name>ep-admin</admin_name> <location>Counter01</location> </data> </admin_info>"\0"</pre>	
<pre><open_device> <device_id>keyboard01</device_id> <data> <type>type_keyboard</type> <buffer>true</buffer> </data> </open_device>"\0"</pre>	➡		Device open
	⬅	<pre><open_device> <device_id>keyboard01</device_id> <code>OK</code> <data_id>1</data_id> </open_device>"\0"</pre>	

Applications		ePOS-Device	Description
<pre> <device_data> <device_id>keyboard01</device_id> <data> <type>setprefix</type> <keycodes array="true">49 </keycodes> <keycodes>50</keycodes> <keycodes>51</keycodes> </data> </device_data>"\0" </pre>	➔		Device control
	➔	<pre> <device_data> <device_id>keyboard01</device_id> <data> <type>onkeypress</type> <keycode>49</keycode> <ascii>a</ascii> </data> <data_id>2</data_id> </device_data>"\0" <device_data> <device_id>keyboard01</device_id> <data> <type>onstring</type> <input>11223344556677</input> <prefix>a</prefix> </data> <data_id>3</data_id> </device_data>"\0" </pre>	Device control
		Key event (data_id=4) occurred	Detect a disconnection
	➔	<pre> <connect><data> <data> <client_id>sock3514555411 </client_id> <protocol_version>2 </protocol_version> </data></data></connect>"\0" </pre>	Begin reconnection
<pre> <data> <old_client_id>sock3514555410 </old_client_id> <new_client_id>sock3514555411 </new_client_id> <received_id>3</received_id> </data> </reconnect> </pre>	➔		
	➔	<pre> <reconnect> <code>OK</code> </reconnect>"\0" <device_data> <device_id>keyboard01</device_id> <data> <type>onkeypress</type> <keycode>50</keycode> <ascii>b</ascii> </data> <data_id>4</data_id> </device_data>"\0" </pre>	

Applications		ePOS-Device	Description
<pre><close_device> <device_id>keyboard01</device_id> </close_device>"\0"</pre>	➡		Device close
	⬅	<pre><close_device> <device_id>keyboard01</device_id> <code>OK</code> <data_id>5</data_id> </close_device>"\0"</pre>	
<pre><disconnect> <data> <client_id>sock3514555411 </client_id> </data> </disconnect></pre>	➡		Disconnection of connection

Communication Box Message Example

Application A		ePOS-Device		Application B
• Opens the Communication Box				
<pre><open_commbox> <sequence>1</sequence> <data> <box_id> box1 </box_id> <member_id> member1 </member_id> </data> </open_commbox> "\0"</pre>	➔			
	➔	<pre><open_commbox> <sequence>1</sequence> <data> <box_id>box1</box_id> <code>OK</code> </data> <data_id>1</data_id> </open_commbox> "\0"</pre>		
			➔	<pre><open_commbox> <sequence>1</sequence> <data> <box_id> box1 </box_id> <member_id> member2 </member_id> </data> </open_commbox> "\0"</pre>
		<pre><open_commbox> <sequence>1</sequence> <data> <box_id>box1</box_id> <code>OK</code> </data> <data_id>1</data_id> </open_commbox> "\0"</pre>	➔	

Application A		ePOS-Device		Application B
• Forwards data to the Communication Box				
<pre> <commbox_data> <sequence>2</sequence> <data> <type>send</type> <box_id>box1</box_id> <message> send_message </message> </data> </commbox_data>"\0" </pre>	➔			
		<pre> <commbox_data> <sequence>2</sequence> <data> <type>onreceive</type> <box_id>box1</box_id> <sender_id> member1 </sender_id> <receiver_id> </receiver_id> <message> send_message </message> </data> <data_id>2</data_id> </commbox_data>"\0" </pre>	➔	
	➔	<pre> <commbox_data> <sequence>2</sequence> <data> <type>send</type> <box_id>box1</box_id> <code>OK</code> <count>1</count> </data> <data_id>2</data_id> </commbox_data>"\0" </pre>		

Application A		ePOS-Device		Application B
• Acquires the transfer history of the Communication Box				
			←	<pre><commbox_data> <sequence>2</sequence> <data> <type> getcommhistory </type> <box_id>box1</box_id> </data> </commbox_data>"\0"</pre>
		<pre><commbox_data> <sequence>2</sequence> <data> <type> getcommhistory </type> <box_id>box1</box_id> <code>OK</code> <history_list> <senderId> member1 </senderId> <receiverId> </receiverId> <message> send_message </message> </history_list> </data> <data_id>3</data_id> </commbox_data>"\0"</pre>	→	
• Closes the Communication Box				
<pre><close_commbox> <sequence>3</sequence> <data> <box_id>box1</box_id> </data> </close_commbox>"\0"</pre>	→			
	←	<pre><close_commbox> <sequence>3</sequence> <data> <box_id>box1</box_id> <code>OK</code> </data> <data_id>3</data_id> </close_commbox>"\0"</pre>		
			←	<pre><close_commbox> <sequence>3</sequence> <data> <box_id>box1</box_id> </data> </close_commbox>"\0"</pre>
		<pre><close_commbox> <sequence>3</sequence> <data> <box_id>box1</box_id> <code>OK</code> </data> <data_id>4</data_id> </close_commbox>"\0"</pre>	→	

Array of <data> elements in <device_data> message

The following programming method is used to describe an array in the <data> tag:

- ❑ Specify the attribute array="true" at the beginning of the element to be used for the array.
- ❑ After that, specify a value with the same element name.

Example: To express "keycodes=(49, 50, 51, 52);"

```
<data>
  <type>setprefix</type>
  <keycodes array="true">49</keycodes>
  <keycodes>50</keycodes>
  <keycodes>51</keycodes>
  <keycodes>52</keycodes>
</data>
```

ePOS-Device XML

This chapter explains the ePOS-Device XML.

ePOS-Device XML List

ePOS-Device XML includes the following.

❑ Message(p. 47)

This is an XML for request messages from the application to the TM-Intelligent and response messages from the TM-Intelligent to the application.

❑ Message Data for Communication Box (p. 48)

(in ePOS-Device Service Ver.2.5 and later)

This XML is used for data communication between applications.

This forms the sub-element for the <data> element in <commbox_data> messages.

❑ Message Data for Each Device (p. 48)

This XML stores the data for requests and responses for each controlled device.

This forms the sub-element for the <data> element in <device_data> messages.

Message

Message		Description	Page
Communication path	<connect>	Establishes a communication path with ePOS-Device Service I/F	51
	<reconnect>	Re-establish communication route with the ePOS-Device Service I/F	52
	<disconnect>	Cut off communication route with ePOS-Device Service I/F	53
Administrator information	<admin_info>	Acquires administrator and installation location information	54
Communication Box *	<open_commbox>	Opens the Communication Box	56
	<close_commbox>	Closes the Communication Box	58
	<commbox_data>	Communication between applications	59
Device open / close	<open_device>	Opens communication with a device	60
	<close_device>	Closes communication with a device	62
Send device control data	<device_data>	Requests device control	64
	<device_data>	Responds to device control	
Error notification	<error>	Notifies you of error contents when an error occurs	65

* ePOS-Device Service Ver. 2.5 or later versions supported

Message Data for Communication Box

Designates the data for data communication between applications with the sub-element for the <data> element for the <commbox_data> message.

Before designating data, this designates the type of message data with the <type> element, then designates the sub-element data. The <type> elements that can be used are listed in the chart below.

(in ePOS-Device Service Ver.2.5 and later)

Value for <type> element	Description	Page
getcommhistory	Acquires the transfer history of the Communication Box. <Request>/<Response>	66
send	Sends data to the Communication Box. <Request>	68
onreceive	Notifies data from the Communication Box. <Response>	69

Message Data for Each Device

Designates the data to control the device with the sub-element of the <data> element for the <device_data> message.

The component elements for message data vary based on the device type.

Before designating the data, designates the message data type with the <type> element, and then designates the sub-element data. The device types and the <type> elements that can be used with each TM intelligent printer are as in the table below.

Device type	Value for <type> element	Description	TM-DT Series	TM-i Series	Page
Device Hub Terminal	shutdown *1	<Request> Shuts down the TM-DT	✓	-	70
	onshutdown *1	<Response> Shutdown result of the TM-DT	✓	-	70
	restart *2	<Request> Restart the TM-DT	✓	-	71
	onrestart *2	<Response> Restart result of the TM-DT	✓	-	71
Barcode Scanner	ondata	<Response> Scan data from a barcode scanner	✓	✓	72
Printer	print	<Request> Print data and setting data to printers	✓	✓	73
	onxmlresult	<Response> Data from printers	✓	✓	74

Device type	Value for <type> element	Description	TM-DT Series	TM-i Series	Page
Hybrid Printer * ¹	lock	<Request> Locks the device port	✓	-	77
	unlock	<Request> Unlocks the device port	✓	-	77
	print	<Request> Print data and setting data to receipt printers	✓	-	78
	onxmlresult	<Response> Data from receipt printers	✓	-	78
	slipprint	<Request> Print data and setting data to slip printers	✓	-	79
	slipcancel	<Request> Cancels the paper insert wait status during slip printing	✓	-	79
	endorseprint	<Request> Print data and setting data to slip printers	✓	-	80
	endorsecancel	<Request> Cancels the paper insert wait status during endorse printing	✓	-	80
	micread	<Request> MICR reading	✓	-	81
	micrcancel	<Request> Cancels the paper insert wait status during MICR reading	✓	-	82
	micrcleaning	<Request> Cleans the MICR mechanism	✓	-	82
	eject * ²	<Request> Ejects the check paper	✓	-	82
	onreceive	<Response> Hybrid printer processing result	✓	-	83
Customer Display	display	<Request> Display data and setting data to the customer display	✓	✓	128
	onxmlresult	<Response> Data from the customer display	✓	✓	129

Device type	Value for <type> element	Description	TM-DT Series	TM-i Series	Page
POS Keyboard	onkeypress	<Response> Input character strings from the POS keyboard	✓	✓	140
	setprefix	<Request> Designates the key code to determine the start of character strings to detect input from the POS keyboard	✓	✓	141
	onstring	<Response> Input character strings from the POS keyboard	✓	✓	142
	setMSRPrefix	<Request> Designates the keycode for determining the card information reception with the keyboard with an MSR	✓	✓	143
	ondata	<Response> Card information from a keyboard with an MSR	✓	✓	144
Serial Device ^{*3}	sendcommand	<Request> Sends commands to the serial device	✓	✓	145
	oncommandreply	<Response> Data from the serial device	✓	✓	146

*1 ePOS-Device Service Ver. 2.5 or later versions supported.

*2 ePOS-Device Service Ver. 3.0 or later versions supported.

*3 Serial devices cannot be used with a TM-i Series printer with the specification of "without a Serial port".

Messages

<connect>

Response

Returns to the application that the connection was established.

Sub-element	Data type	Description
data	string	Connection information For details, refer to the data below. (Returned only when the connection is successful.)



Send the following message with the application, triggered by receiving this message.

❑ data

Sub-element	Data type	Description
client_id	string	ID of the connected client
protocol_version	string	Version of the connected TM-Intelligent

❑ Example

```
<connect>
  <data>
    <client_id>1234567890</client_id>
    <protocol_version>2.0</protocol_version>
  </data>
</connect>
```

<reconnect>

Notifies the confirmation of reconnection for network communication with the TM-Intelligent.

Request

Requests confirmation of reconnection to the TM-Intelligent.

Sub-element	Data type	Description
old_client_id	string	Specifies the client_id received by <connect> from the previous session.
new_client_id	string	Specifies the client_id received by <connect> from the current session.
received_id	int	Specifies what last received communication with the data_id received from all communication data

❑ Example

```
<reconnect>
  <data>
    <new_client_id>1234567890</new_client_id>
    <old_client_id>0987654321</old_client_id>
    <received_id>112</received_id>
  </data>
</reconnect>
```

Response

Returns the confirmation result for reconnection to the application.

Sub-element	Data type	Description
code	string	Confirmation result for reconnection. For details, refer to the code below.

❑ code

Element value	Description
"OK"	Successfully reconnected
"CLIENT_NOT_FOUND"	A connection that corresponds to the specified client_id does not exist
"PARAM_ERROR"	Parameter error occurred
"SYSTEM_ERROR"	System error occurred

❑ Example

Normal

```
<reconnect>
  <code>OK</code>
</reconnect>
```

Error

```
<reconnect>
  <code>"PARAM_ERROR"</code>
</reconnect>
```

<disconnect>

Closes the TCP connection with the TM-Intelligent.

Request

Requests that the TM-Intelligent close the connection.

Sub-element	Data type	Description
data	string	Client information to close the connection. For details, refer to the data below.

❑ data

Sub-element	Data type	Description
client_id	string	Specifies the client_id to close the connection from the TM-Intelligent.

❑ Example

```
<disconnect>
  <data>
    <client_id>1234567890</client_id>
  </data>
</disconnect>
```

Response

Returns the TCP connection disconnection result to the application.

Sub-element	Data type	Description
code	string	Confirmation result for reconfirmation. For details, refer to the code below.

❑ code

Element value	Description
"OK"	Close the TCP connection
"PARAM_ERROR"	Parameter error occurred
"SYSTEM_ERROR"	System error occurred

❑ Example

Normal

```
<disconnect>
  <code>OK</code>
</disconnect>
```

Error

```
<disconnect>
  <code>"PARAM_ERROR"</code>
</disconnect>
```

<admin_info>

Returns the administrator information set in the TM-Intelligent.

The information is set with EPSON TMNet WebConfig.

Request

Requests the administrator information from the TM-Intelligent.

□ Example

```
<admin_info></admin_info>
```

Response

Returns the administrator information to the application.

Sub-element	Data type	Description
code	string	Result of acquiring the administrator information For details, refer to the code below.
data	string	Administrator information Only returned if successfully acquired. For details, refer to the data below.

□ code

Element value	Description
"OK"	Successfully acquired
"PARAM_ERROR"	Parameter error occurred
"SYSTEM_ERROR"	System error occurred

□ data

Sub-element	Data type	Description
admin_name	string	Administrator information
location	string	Installation location information

Character strings in received data

Escape processing is performed for the following character strings.

Character string before replacement	After replacement
& (Ampersand)	&
> (Greater-than symbol)	>
< (Lesser-than symbol)	<
"(Double quotation mark)	"
' (Single quotation mark)	'

□ Example

Normal

```
<admin_info>
  <code>OK</code>
  <data>
    <admin_name>Administrator A</admin_name>
    <location>Counter1&Counter2</location>
  </data>
</admin_info>
```

Error

```
<admin_info>
  <code>"PARAM_ERROR"</code>
</admin_info>
```

<open_commbox>

Opens the Communication Box and enables communication between applications.
(in ePOS-Device Service Ver.2.5 and later)

Request

Requests opening of the Communication Box.

Sub-element	Data type	Description
sequence	string	Sequence number
data	string	Information of the Communication Box to open. For details, refer to the data below.

□ data

Sub-element	Data type	Description
box_id	string	Identifier of the Communication Box to open.
member_id	string	Member ID for identifying the client (application) in the Communication Box.

□ Example

```
<open_commbox>
<sequence>1</sequence>
<data>
  <box_id>box1</box_id>
  <member_id>member1</member_id>
</data>
</open_commbox>
```

Response

Returns the Communication Box open result to the application.

Sub-element	Data type	Description
sequence	string	The sequence number designated with the request.
data	string	Information of the Communication Box open result. For details, refer to the data below.
data_id	string	ID for identifying the sent data.

□ data

Sub-element	Data type	Description
box_id	string	Identifier of the Communication Box that is opened.
code	string	The processing result. For details, refer to the code below.

• code

Element value	Description
"OK"	Successfully opened the Communication Box.
"ALREADY_OPENED"	The Communication Box is already opened.
"MEMBERID_ALREADY_USED"	The specified member ID is already in use.
"BOX_COUNT_OVER"	The maximum number of Communication Boxes that can be made is exceeded.
"BOX_CLIENT_OVER"	The maximum number that can belong to a Communication Box is exceeded.
"PARAMETER_ERROR"	Parameter error occurred
"SYSTEM_ERROR"	System error occurred

<close_commbox>

Closes the Communication Box. (in ePOS-Device Service Ver.2.5 and later)

Request

Requests closing of the Communication Box.

Sub-element	Data type	Description
sequence	string	Sequence number
data	string	Information of the Communication Box to close. For details, refer to the data below.

□ data

Sub-element	Data type	Description
box_id	string	Identifier of the Communication Box to close.

□ Example

```
<close_commbox>
<sequence>2</sequence>
<data>
  <box_id>box1</box_id>
</data>
</close_commbox>
```

Response

Returns the Communication Box close result to the application.

Sub-element	Data type	Description
sequence	string	The sequence number designated with the request.
data	string	Information of the Communication Box close result. For details, refer to the data below.
data_id	string	ID for identifying the sent data.

□ data

Sub-element	Data type	Description
box_id	string	Identifier of the Communication Box that is closed.
code	string	The processing result. For details, refer to the code below.

• code

Element value	Description
"OK"	Successfully closed the Communication Box.
"NOT_OPENED"	A Communication Box is not open.
"SYSTEM_ERROR"	System error occurred

<commbox_data>

Performs data communication between applications using the Communication Box.
(in ePOS-Device Service Ver.2.5 and later)

Request

Communicates with the Communication Box.

Sub-element	Data type	Description
sequence	string	Sequence number
data	string	The data used for communication between applications using the Communication Box. Contents are defined for each communication purpose. For details, refer to Message Data for Communication Box (p.48) .

Response

Returns the communication result of the Communication Box and notification of data reception to the application.

Sub-element	Data type	Description
sequence	string	The sequence number designated with the request.
data	string	The data used for communication between applications using the Communication Box. Contents are defined for each communication purpose. For details, refer to Message Data for Communication Box (p.48) .
data_id	int	ID for identifying the sent data.

<open_device>

Makes the device linked to the device ID usable. Provides exclusive use for the designated device to the requesting application (except for printers).

Request

Designates the device to open.

Sub-element	Data type	Description
device_id	string	The device ID designated with EPSON TMNet WebConfig. The customer display is "local_display".
data	string	For details, refer to the data below.

□ data

Sub-element	Data type	Description
type	string	Specifies the device category name. • "type_dt": Device hub terminal (TM-DT) * • "type_scanner": Barcode scanner • "type_keyboard": POS keyboard • "type_printer": Printer • "type_hybrid_printer": Hybrid Printer * • "type_display": Customer display • "type_simple_serial": Serial communication device
buffer	string	Specifies the buffer function for when a connection is disconnected. • "true" :Enable • "false" :Disable

* ePOS-Device Service Ver. 2.5 or later versions supported



- With TM-i, the following devices are not supported:
 - * "type_dt"
 - * "type_hybrid_printer"
- A TM-i Series printer with the specification of "without a Serial port" does not support "type_simple_serial".

□ Example

```
<open_device>
  <device_id>scanner_001</device_id>
  <data>
    <type>type_scanner</type>
  </data>
</open_device>
```

Response

Returns the device open result to the application.

Sub-element	Data type	Description
device_id	string	The device ID designated with the request.
code	string	The processing result. For details, refer to the code below.
data_id	int	If "true" is designated for <buffer> of the <open_device> message, the sent data ID is set. If "false" is designated, a space is set.

❑ code

Element value	Description
"OK"	Processed successfully
"DEVICE_NOT_FOUND"	The designated device does not exist
"DEVICE_IN_USE"	The designated device is in use
"DEVICE_OPEN_ERROR"	Failed to open the designated device
"DRIVER_ERROR"	An error occurred in the device control script
"SYSTEM_ERROR"	System error occurred
"DEVICE_TYPE_INVALID"	The designated device type is invalid
"PARAM_ERROR"	Parameter error occurred

❑ Example

Normal

```
<open_device>
  <device_id>scanner_001</device_id>
  <code>OK</code>
  <data_id>2</data_id>
</open_device>
```

Error

```
<open_device>
  <code>DEVICE_NOT_FOUND</code>
  <device_id>scanner_001</device_id>
  <data_id>2</data_id>
</open_device>
```

<close_device>

Closes the opened device.

Request

Designates the device to close.

Sub-element	Data type	Description
device_id	string	The device ID designated with EPSON TMNet WebConfig. The customer display is "local_display".

□ Example

```
<close_device>
  <device_id>local_display</device_id>
</close_device>
```

Response

Returns the device close result to the application.

Sub-element	Data type	Description
device_id	string	The device ID designated with the request.
code	string	The processing result. For details, refer to the code below.
data_id	int	If "true" is designated for <buffer> of the <open_device> message, the sent data ID is set. If "false" is designated, a space is set.

code

Element value	Description
"OK"	Processed successfully
"DEVICE_NOT_FOUND"	The designated device does not exist
"DEVICE_NOT_OPEN"	The designated device ID is not open
"DEVICE_CLOSE_ERROR"	Failed to close the designated device
"SYSTEM_ERROR"	System error occurred
"PARAM_ERROR"	Parameter error occurred

Example

Normal

```
<close_device>
  <device_id>local_display</device_id>
  <code>OK</code>
  <data_id>2</data_id>
</close_device>
```

Error

```
<close_device>
  <device_id>scanner_001</device_id>
  <code>DEVICE_NOT_FOUND</code>
  <data_id>2</data_id>
</close_device>
```

<device_data>

Request

Sends data to a device. Includes device control commands and print / display data.

Sub-element	Data type	Description
sequence	string	An arbitrary value to specify request messages during sending errors. If you designate a request message, it will be set in the sequence of the error message during an error.
device_id	string	The opened device ID.
data	string	Data for device control. Contents are defined for each device type. For details, refer to Message Data for Each Device (p.48) .

□ Example

```
<device_data>
  <sequence>123</sequence>
  <device_id>poskeyboard001</device_id>
  <data>
    <type>setprefix</type>
    <keycodes array="true">49</keycodes>
    <keycodes>50</keycodes>
    <keycodes>51</keycodes>
    <keycodes>52</keycodes>
  </data>
</device_data>
```

Response

Returns the data from the device. Includes the result of device control, events which occurred at the device, and input data from the device.

Sub-element	Data type	Description
sequence	string	Fixed as 0
device_id	string	Device ID where the data occurred
data	string	Data that occurred. Contents are defined for each device type. For details, refer to Message Data for Each Device (p.48) .
data_id	int	If "true" is designated for <buffer> of the <open_device> message, the sent data ID is set. If "false" is designated, a space is set.

□ Example

```
<device_data>
  <sequence>0</sequence>
  <device_id>poskeyboard001</device_id>
  <data>
    <type>onstring</type>
    <input>&#2398749238429</input>
    <prefix>49</prefix>
  </data>
  <data_id>2</data_id>
</device_data>
```

<error>**Response**

Returns the device control command and other common errors.

Sub-element	Data type	Description
sequence	string	ID to specify the message that caused the error. It is the same value as the sequence designated by the request.
device_id	string	Device ID where the error occurred. Designated if an error is in the device control message.
code	string	Displays the error that occurred using a code. For details, refer to the following.
data	string	The details of errors based on the driver are designated when a "DRIVER_ERROR" occurs.
data_id	int	If "true" is designated for <buffer> of the <open_device> message, the sent data ID is set. If "false" is designated, a space is set.

❑ code

- <When there is an error in <device_data>

Element value	Description
"DEVICE_NOT_FOUND"	The designated device ID does not exist
"DEVICE_NOT_OPEN"	The designated device is not open
"SEND_ERROR"	An error occurred sending the data to the physical device
"DRIVER_ERROR"	Detected an error in the device
"PARAM_ERROR"	Parameter error occurred
"FUNCTION_NOT_FOUND"	The designated function does not exist in the driver

- Any other error

Element value	Description
"SYSTEM_ERROR"	System error occurred

❑ Example

```
<error>
  <sequence>123</sequence>
  <device_id>scanner002</device_id>
  <code>DEVICE_NOT_OPEN</code>
  <data></data>
  <data_id>2</data_id>
</error>
```

Message Data for Communication Box



Available with ePOS-Device Service Ver.2.5 or later.

<type>getcommhistory</type>

Acquires the transfer history of the Communication Box.

Request

Requests acquisition of transfer history of the Communication Box to the ePOS-Device Service I/F.

Sub-element	Data type	Description
box_id	string	Identifier of the Communication Box to acquire transfer history from
all_history *	boolean	Specify whether to acquire the entire transmission history of the communication box. For details, refer to the history_list below.

* ePOS-Device Service Ver. 3.0 or later versions supported.

□ history_list

Element value	Description
"true" / "1"	Acquires the entire transmission history of the communication box.
"false" / "0"	Acquires the transmission history only for data sent to the own member ID. If omitted, "false"/"0" is specified.

□ Example

```
<commbox_data>
<sequence>4</sequence>
<data>
  <type>getcommhistory</type>
  <box_id>box1</box_id>
</data>
</commbox_data>
```

Response

Returns the transfer history of the Communication Box.

Sub-element	Data type	Description
box_id	string	Identifier of the Communication Box whose transfer history was acquired
code	string	Result of the sent data. For details, refer to the code below.
history_list	string	Acquired transfer history. Notifies <history_list> the same number of times as the number of the acquired transfer history items. Stored in the latest order of data. For details, refer to the history_list below.

code

Element value	Description
"OK"	Successfully acquired transfer history of the Communication Box.
"NOT_OPENED"	A Communication Box is not open.
"SYSTEM_ERROR"	System error occurred

history_list

Element value	Description
senderId	Sender's member ID
receiverId	Receiver's member ID
message	Received message

Example

```
<commbox_data>
  <sequence>4</sequence>
  <data>
    <type>getcommhistory</type>
    <box_id>box1</box_id>
    <code>OK</code>
    <history_list>
      <senderId>member1</senderId>
      <receiverId></receiverId>
      <message>send_message3</message>
    </history_list>
    <history_list>
      <senderId>member1</senderId>
      <receiverId></receiverId>
      <message>send_message2</message>
    </history_list>
    <history_list>
      <senderId>member1</senderId>
      <receiverId></receiverId>
      <message>send_message</message>
    </history_list>
  </data>
  <data_id>10</data_id>
</commbox_data>
```

<type>send</type>

Forwards data to the Communication Box.

Request

Forwards messages to the Communication Box.

Sub-element	Data type	Description
box_id	string	Identifier of the Communication Box to forward the message to.
message	string	Message to forward.
member_id	string	Member ID to forward. To forward message to all the clients (applications) belonging to the Communication Box, specify " "(space).

❏ Example

```
<commbox_data>
<sequence>3</sequence>
<data>
<type>send</type>
<box_id>box1</box_id>
<message>send_message</message>
</data>
</commbox_data>
```

Response

Returns the message transfer result.

Sub-element	Data type	Description
box_id	string	Identifier of the Communication Box that forwarded the message
code	string	Forward result of the message. For details, refer to the code below.
count	string	The number of clients that attempted to forward a message.

❏ code

Element value	Description
"OK"	Successfully forwarded the message.
"NOT_OPENED"	A Communication Box is not open.
"MEMBER_NOT_FOUND"	The specified member ID cannot be found.
"SYSTEM_ERROR"	System error occurred

<type>onreceive</type>

Acquires the data forwarded to the Communication Box.

Response

Returns the message transfer result.

Sub-element	Data type	Description
box_id	string	Identifier of the Communication Box that forwarded the message.
sender_id	string	Member ID that forwarded the message.
receiver_id	string	Member ID that acquires the forwarded message. (When all the clients (applications) belonging to the Communication Box acquire the message, " "(space) is returned.)
message	string	Message forwarded to the Communication Box.

□ Example

```
<commbox_data>
  <sequence>3</sequence>
  <data>
    <type>onreceive</type>
    <box_id>box1</box_id>
    <sender_id>member1</sender_id>
    <receiver_id></receiver_id>
    <message>send_message</message>
  </data>
  <data_id>19</data_id>
</commbox_data>
```

Message Data for Device Hub Terminal



- Cannot be used with TM-i Series.
- Available with ePOS-Device Service Ver.2.5 or later.
- If sending this message data, use EPSON TMNet WebConfig to perform shutdown settings for the TM-DT.
([Settings]-[System settings]-[Shutdown settings]-[ePOS-Device])

<type>shutdown</type>

Request

Shuts down the power source of the device hub terminal (TM-DT).

Sub-element	Data type	Description
password	string	Specifies the password for EPSON TMNet WebConfig.

❑ Example

```
<device_data>
<device_id>local_dt</device_id>
<data>
  <type>shutdown</type>
  <password>epson</password>
</data>
</device_data>
```

<type>onshutdown</type>

Response

Returns the shutdown results of the device hub terminal (TM-DT) to the application.

Sub-element	Data type	Description
status	string	Result of the sent data. For details, refer to the status below.

❑ status

Element value	Description
"SUCCESS"	Shutdown successfully completed
"AUTHENTICATION_ERROR"	Password authentication error
"ACCESS_DENIED"	Shutdown from ePOS-Device XML is not set
"SYSTEM_ERROR"	System error occurred

<type>restart</type>

ePOS-Device Service Ver.3.0 or later versions supported.

Request

Restart the power source of the device hub terminal (TM-DT).

Sub-element	Data type	Description
password	string	Specifies the password for EPSON TMNet WebConfig.

□ Example

```
<device_data>
  <device_id>local_dt</device_id>
  <data>
    <type>restart</type>
    <password>epson</password>
  </data>
</device_data>
```

<type>onrestart</type>

ePOS-Device Service Ver.3.0 or later versions supported.

Response

Returns the restart results of the device hub terminal (TM-DT) to the application.

Sub-element	Data type	Description
status	string	Result of the sent data. For details, refer to the status below.

□ status

Element value	Description
"SUCCESS"	Succeeded
"AUTHENTICATION_ERROR"	Password authentication error
"ACCESS_DENIED"	Shutdown from ePOS-Device XML is not set
"SYSTEM_ERROR"	System error occurred

Message Data for the Barcode Scanner

<type>ondata</type>

Response

Returns the scan data from the barcode scanner to the application.

Sub-element	Data type	Description
input	string	Scan data string received.

Character strings in received data

Escape processing is performed for the following character strings.

Character string before replacement	After replacement
& (Ampersand)	&
> (Greater-than symbol)	>
< (Lesser-than symbol)	<
"(Double quotation mark)	"
' (Single quotation mark)	'

❏ Example

```
<data>
<type>ondata</type>
<input>F12345678909123</input>
</data>
```

Message Data for Printers



For print data, refer to [Printer Control XML \(p.85\)](#).

<type>print</type>

Request

Sends the print data and setting data to printers.

Sub-element	Data type	Description
timeout	int	Sets the time out limit for requests in milliseconds. 1,000 to 300,000 (whole numbers)
printjobid *	string	Sets the print job ID. This element can be omitted. 1 to 30 alphanumeric characters, including underscores, hyphens and periods, may be used. For a programming example, refer to Print Job ID (p.75) .
printdata	string	Designates the data to print. For details, refer to Printer Control XML (p.85) .

* ePOS-Device Service Ver. 2.6 or later versions supported

Example

```
<data>
<type>print</type>
<timeout>10000</timeout>
<printdata>
<!-- Printer Control XML -->
  <epos-print xmlns=
    .
  </epos-print>
</printdata>
</data>
```

<type>onxmlresult</type>

Response

Returns the print result to the application.

Sub-element	Data type	Description
printjobid *	string	Print job ID. For details, refer to Specifying the Print Job ID from the Application (p.17) . For a programming example, refer to Print Job ID (p.75) .
resultdata	string	Result of the sent data. For details, refer to <response> (p.87) .

* ePOS-Device Service Ver. 2.6 or later versions supported

❏ Example

```
<data>
  <type>onxmlresult</type>
  <resultdata>
    <response success="true"/>
  </resultdata>
</data>
```

Print Job ID

Request

Specifies the print job ID for the print data.

- ❑ To specify a print job ID from the application

```
<device_data>
  <device_id>local_printer</device_id>
  <data>
    <type>print</type>
    <timeout>60000</timeout>
    <printjobid>ABC123</printjobid>
    <printdata>
      <epos-print xmlns="http://www.epson-pos.com/schemas/2011/03/epos-print">
        <text>Hello, World!&#10;</text>
        <cut />
      </epos-print>
    </printdata>
  </data>
</device_data>
```

- ❑ To issue a job ID from the spooler

```
<device_data>
  <device_id>local_printer</device_id>
  <data>
    <type>print</type>
    <timeout>60000</timeout>
    <printdata>
      <epos-print xmlns="http://www.epson-pos.com/schemas/2011/03/epos-print">
        <text>Hello, World!&#10;</text>
        <cut />
      </epos-print>
    </printdata>
  </data>
</device_data>
```

- ❑ To acquire print job status from the spooler

```
<device_data>
  <device_id>local_printer</device_id>
  <data>
    <type>print</type>
    <printjobid>123</printjobid>
    <timeout>60000</timeout>
    <printdata>
      <epos-print xmlns="http://www.epson-pos.com/schemas/2011/03/epos-print" />
    </printdata>
  </data>
</device_data>
```



Empty print data is sent when requesting the acquisition of the print job status.
As such, child elements of the <epos-print> element are not specified.

Response

The print job ID for the requested process is returned to the application.

❑ To specify a print job ID from the application

```
<device_data>
  <device_id>local_printer</device_id>
  <data>
    <type>onxmlresult</type>
    <printjobid>123</printjobid>
    <resultdata>
      <response xmlns="http://www.epson-pos.com/schemas/2011/03/epos-print"
        success="true" code="" status="252641302" battery="0" />
    </resultdata>
  </data>
</device_data>
```

❑ To issue a job ID from the spooler

```
<device_data>
  <device_id>local_printer</device_id>
  <data>
    <type>onxmlresult</type>
    <printjobid>123</printjobid>
    <resultdata>
      <response xmlns="http://www.epson-pos.com/schemas/2011/03/epos-print"
        success="true" code="" status="2" battery="0" />
    </resultdata>
  </data>
</device_data>
```



- The TM intelligent printer will issue a print job ID if the print job ID is not specified by the application.
- As the printing process has not been performed at this time, success will be set to "true," code to "" (left empty), status to "0x00000002" or "0x80000000" and battery to "0".

❑ To acquire print job status from the spooler

```
<device_data>
  <device_id>local_printer</device_id>
  <data>
    <type>onxmlresult</type>
    <printjobid>ABC123</printjobid>
    <resultdata>
      <response xmlns="http://www.epson-pos.com/schemas/2011/03/epos-print"
        success="true" code="" status="252641302" battery="0" />
    </resultdata>
  </data>
</device_data>
```

Message Data for Hybrid Printers



- Cannot be used with TM-i Series.
- Available with ePOS-Device Service Ver.2.5 or later.

<type>lock</type>

Request

Locks the device port. Execute when performing MICR reading or you want to lock the device.



If you locked a port with this message data, be sure to send the [<type>unlock</type> \(p.77\)](#) message before closing the application.

If you close the application without sending the unlock, the next port cannot be locked for about five minutes until the ePOS-Device Service I/F unlocks the port.

The [<type>onreceive</type> \(p.83\)](#) message returns 'ERROR_DEVICE_BUSY'.

Example

```
<device_data>
  <device_id>local_printer</device_id>
  <data>
    <type>lock</type>
  </data>
</device_data>
```

<type>unlock</type>

Request

Unlocks the device port.

Example

```
<device_data>
  <device_id>local_printer</device_id>
  <data>
    <type>unlock</type>
  </data>
</device_data>
```

<type>print</type>

Request

Sends the print data and setting data to the receipt printer.

Sub-element	Data type	Description
timeout	int	Sets the time out limit for requests in milliseconds. 1,000 to 60,000 (whole numbers)
printdata	string	Designates the data to print. For details, refer to Printer Control XML (p.85) .

❑ Example

```
<device_data>
  <device_id>local_printer</device_id>
  <data>
    <type>print</type>
    <timeout>10000</timeout>
    <printdata>
      <!-- Printer Control XML -->
      <epos-print xmlns=
        .
      </epos-print>
    </printdata>
  </data>
</device_data>
```

<type>onxmlresult</type>

Response

Returns the receipt print result to the application.

Sub-element	Data type	Description
resultdata	string	Result of the sent data. For details, refer to <response> (p.87) .

❑ Example

```
<data>
  <type>onxmlresult</type>
  <resultdata>
    <response success=    />
  </resultdata>
</data>
```

<type>slipprint</type>**Request**

Sends the print data and setting data to the slip printer.

Sub-element	Data type	Description
timeout	int	Specifies the timeout time (ms) for the paper insert wait status. 60,000 to 900,000 (whole numbers)
printdata	string	Designates the data to print. For details, refer to Printer Control XML (p.85) .

□ Example

```
<device_data>
  <device_id>local_printer</device_id>
  <data>
    <type>slipprint</type>
    <timeout>60000</timeout>
    <printdata>
      <!-- Printer Control XML -->
      <epos-print xmlns=
        .
      </epos-print>
    </printdata>
  </data>
</device_data>
```

<type>slipcancel</type>**Request**

Cancels the paper insert wait status during slip printing.

□ Example

```
<device_data>
  <device_id>local_printer</device_id>
  <data>
    <type>slipcancel</type>
  </data>
</device_data>
```

<type>endorseprint</type>

Request

Sends the print data and setting data to the endorse printer.

Sub-element	Data type	Description
40cplmode	boolean	Sets the 40cpl mode. For details, refer to the 40cplmode below.
timeout	int	Specifies the timeout time (ms) for the paper insert wait status. 60,000 to 900,000 (whole numbers)
printdata	string	Designates the data to print. For details, refer to Printer Control XML (p.85) .

❑ 40cplmode

Element value	Description
true	Enables the 40cpl mode.
false	Disables the 40cpl mode.

❑ Example

```
<device_data>
<device_id>local_printer</device_id>
<data>
  <type>endorseprint</type>
  <40cplmode>>false</40cplmode>
  <timeout>60000</timeout>
  <printdata>
    <!-- Printer Control XML -->
    <epos-print xmlns=
      .
    </epos-print>
  </printdata>
</data>
</device_data>
```

<type>endorsecancel</type>

Request

Cancels the paper insert wait status during endorse printing.

❑ Example

```
<device_data>
<device_id>local_printer</device_id>
<data>
  <type>endorsecancel</type>
</data>
</device_data>
```

<type>micrread</type>**Request**

Executes MICR reading.

If you set the paper after executing this message, MICR reading is performed.

Sub-element	Data type	Description
ignoreerror	boolean	Specifies the operation to perform when unreadable characters are detected. For details, refer to the ignoreerror below.
font	string	Specifies the MICR font to read. For details, refer to the font below.
timeout	int	Specifies the timeout time (ms) for the paper insert wait status. 60,000 to 900,000 (whole numbers)

❑ ignoreerror

Element value	Description
true	Replaces unreadable characters with '?' and continues processing.
false	Stops processing.

❑ font

Element value	Description
"FONT_E13B"	Reads the E13B font.
"FONT_CMC7"	Reads the CMC7 font.

❑ Example

```
<device_data>
  <device_id>local_printer</device_id>
  <data>
    <type>micrread</type>
    <ignoreerror>true</ignoreerror>
    <font>MICR_E13B</font>
    <timeout>60000</timeout>
  </data>
</device_data>
```

<type>micrcleaning</type>

Request

Cleans the MICR mechanism.

After executing this message, set the cleaning sheet to clean the MICR mechanism.

Sub-element	Data type	Description
timeout	int	Specifies the timeout time (ms) for the paper insert wait status. 60,000 to 900,000 (whole numbers)

❑ Example

```
<device_data>
  <device_id>local_printer</device_id>
  <data>
    <type>micrcleaning</type>
    <timeout>123456</timeout>
  </data>
</device_data>
```

<type>micrcancel</type>

Request

Cancels the paper insert wait status during MICR reading or MICR mechanism cleaning.

❑ Example

```
<device_data>
  <device_id>local_printer</device_id>
  <data>
    <type>micrcancel</type>
  </data>
</device_data>
```

<type>eject</type>

Request

Ejects the check paper.

❑ Example

```
<device_data>
  <device_id>local_printer</device_id>
  <data>
    <type>eject</type>
  </data>
</device_data>
```

<type>onreceive</type>

Acquires the processing result of the HybridPrinter object.

Response

Returns the message transfer result.

Sub-element	Data type	Description
eventtype	string	Method name that returns an event
success	string	Processing result of the method that returns an event. For details, refer to the success below.
code	string	Execution result of the method that returns an event. For details, refer to the code below.
status	string	Hybrid printer status. For details, refer to the status below.
data	string	Receives MICR reading data when the method that returns an event is "micread". Receives " "(space) when the method is not "micread".

❑ success

Element value	Description
"true" or "1"	Succeeded.
"false" or "0"	Failed.

❑ code

Element value	Description
'SUCCESS'	Succeeded
'EPTR_AUTOMATICAL'	An automatically recoverable error occurred (During receipt printing only)
'EPTR_COVER_OPEN'	A cover open error occurred (During receipt printing only)
'EPTR_CUTTER'	An autocutter error occurred
'EPTR_MECHANICAL'	A mechanical error occurred
'EPTR_REC_EMPTY'	No paper in roll paper end sensor
'EPTR_UNRECOVERABLE'	An unrecoverable error occurred (During receipt printing only)
'SchemaError'	The request document contains a syntax error (During receipt printing only)
'DeviceNotFound'	The printer with the specified device ID does not exist (During receipt printing only)
'PrintSystemError'	An error occurred on the printing system (During receipt printing only)
'EX_BADPORT'	An error was detected on the communication port (During receipt printing only)
'EX_TIMEOUT'	A print timeout occurred (During receipt printing only)
'ERROR_DEVICE_NOT_FOUND'	The device cannot be found

Element value	Description
'ERROR_DEVICE_BUSY'	Failed to open the port
'CANCEL'	Cancels the paper insert wait status
'ERROR_TIMEOUT'	Paper insert wait timeout
'ERROR_PARAMETER'	Parameter error
'ERROR_NOT_SUPPORTED'	The endorse printer is not installed
'EPTR_SCHEMAERROR'	There is an error in the syntax of the request document
'ERROR_COMMAND'	The cancelled processing is not in the paper insert wait status
'EMICR_ILLEGAL_LENGTH'	The inserted paper is not the correct length
'EMICR_NO_MICR'	Did not detect MICR data
'EMICR_RECOGNITION'	Detected unrecognizable MICR text
'EMICR_READ'	Error occurred during MICR reading
'EMICR_NOISE_DETECTED'	Detected noise error
'EMICR_COVER_OPEN'	Cover was opened during MICR reading
'EMICR_PAPER_JAM'	Paper jam error occurred

❑ status

- Depending on the printer situation, an OR function is acquired in a decimal expression from the following values.

Element value	Description
"0x00000001"	No response from the TM printer
"0x00000002"	Printing is successfully completed
"0x00000004"	Status of the 3rd pin of the drawer kick-out connector = "H"
"0x00000008"	Offline
"0x00000020"	The cover is open
"0x00000040"	Paper is being fed by a paper feed switch operation
"0x00000100"	Waiting to be brought back online
"0x00000200"	The paper feed switch is being pressed (ON)
"0x00000400"	A mechanical error occurred
"0x00000800"	An autocutter error occurred
"0x00002000"	An unrecoverable error occurred
"0x00004000"	An automatically recoverable error occurred
"0x00010000"	Slip section is waiting for paper to be inserted
"0x00020000"	No paper in roll paper near end sensor
"0x00040000"	Slip section is waiting for paper to be removed
"0x00080000"	No paper in roll paper end sensor
"0x00200000"	No paper in the TOF detector
"0x00400000"	No paper in the BOF detector
"0x01000000"	A buzzer is on (only for applicable devices)
"0x01000000"	Slip not selected
"0x02000000"	Cannot print on a slip
"0x80000000"	The spooler has stopped (Not used)

Printer Control XML

This chapter explains the XML document that controls printers.



For printer message data, refer to [Message Data for Printers \(p.73\)](#).

<epos-print>

This is the XML document sent from the application to printers. It requests that the printer execute the designated function. <epos-print> contains the element.

Attribute

❑ xmlns

- We're announcing the epos-print namespace. Find the namespace below.
<http://www.epson-pos.com/schemas/2011/03/epos-print>

❑ force

- This is the forced transmission mode. (in ePOS-Device Service Ver.2.2 and later)
If you enable forced transmission mode, print commands are forcibly sent to the printer.

Attribute value	Description
true or 1	Sets forced transmission mode.
false or 0	Sets normal transmission mode.



- Use forced transmission mode when the printer is offline.
It will result in an error if the printer is online.
- The following functions are enabled in forced transmission mode.
 - * Drawer kick-out ([<pulse> \(p.120\)](#))
 - * Stopping the buzzer ([<sound> \(p.121\)](#))
 - * Recovery from errors that can be recovered from ([<recovery> \(p.127\)](#))
 - * Reset ([<reset> \(p.127\)](#))
 - * Sending commands in real time ([<command> \(p.123\)](#))

Sub-element

Sub-element	Function	Receipt/ Label printing	Slip printing	Endorse printing	Endorse printing 40cpl mode	Printer's print mode		Page
						Standard mode	Page mode	
<text>	Prints text	●	●	●	●	●	●	90
<feed>	Paper feed	●	●	●	●	●	●	96
<image>	Prints raster image	●	▲	▲	▲	●	●	98
<logo>	Prints NV logo	●	●	▲	▲	●	●	100
<barcode>	Prints barcode	●	●	▲	▲	●	●	101
<symbol>	Prints symbol	●	▲	▲	▲	●	●	105
<hline>	Prints horizontal line	●	▲	▲	▲	●	▲	110
<vline-begin>	Starts vertical line	●	▲	▲	▲	●	▲	111
<vline-end>	Ends vertical line	●	▲	▲	▲	●	▲	112
<page>	Page mode	●	●	▲	▲	●	▲	113
<area>	Sets a page mode print area	●	▲	▲	▲	▲	●	114
<direction>	Sets a page mode print direction	●	●	▲	▲	▲	●	115
<position>	Sets a page mode print position	●	●	▲	▲	▲	●	116
<line>	Draws a page mode line	●	▲	▲	▲	▲	●	117
<rectangle>	Draws a page mode rectangle	●	▲	▲	▲	▲	●	118
<cut>	Paper cut	●	▲	▲	▲	●	▲	119
<pulse>	Drawer kick-out	●	●	●	●	●	▲	120
<sound>	Sounds a buzzer	●	▲	▲	▲	●	▲	121
<command>	Inserts a command	●	●	●	●	●	●	123
<layout> *	Setting of paper layout	●	▲	▲	▲	●	▲	124
<recovery> *	Recovers from an error	●	●	●	●	●	▲	127
<reset> *	Resets the printer	●	●	●	●	●	▲	127

* ePOS-Device Service Ver. 2.2 or later versions supported

□ Example

```
<!-- Example with a text tag added as a sub-element -->
<epos-print xmlns="http://www.epson-pos.com/schemas/2011/03/epos-print">
  <text>Hello world!!</text>
</epos-print>
```

<response>

This is an XML document that is returned from the printer to the application.

Attribute**❑ success**

- Obtains the print result.

Attribute value	Description
"true" or "1"	- Printing succeeded - Processing succeeded (When the spooler function is enabled) *
"false" or "0"	- Printing Failed - Processing Failed (When the spooler function is enabled) *

* ePOS-Device Service Ver. 2.6 or later versions supported

❑ code

- Obtains the error code.

Attribute value	Description
"EPTR_AUTOMATIC"	An automatically recoverable error occurred
"EPTR_BATTERY_LOW"	No remaining battery
"EPTR_COVER_OPEN"	A cover open error occurred
"EPTR_CUTTER"	An autocutter error occurred
"EPTR_MECHANICAL"	A mechanical error occurred
"EPTR_REC_EMPTY"	No paper in roll paper end sensor
"EPTR_UNRECOVERABLE"	An unrecoverable error occurred
"SchemaError"	The request document contains a syntax error
"DeviceNotFound"	The printer with the specified device ID does not exist
"PrintSystemError"	An error occurred on the printing system
"EX_BADPORT"	An error was detected on the communication port
"EX_TIMEOUT"	A print timeout occurred
"EX_SPOOLER" *	The specified job ID does not exist
"JobNotFound" *	There is not enough space available in the printing queue
"Printing" *	Print job now printing

* ePOS-Device Service Ver. 2.6 or later versions supported

□ status

- Depending on the printer situation, an OR function is acquired in a decimal expression from the following values.

Attribute value	Description
"0x00000001"	No response from the TM printer
"0x00000002"	Printing is successfully completed
"0x00000004"	- Status of the 3rd pin of the drawer kick-out connector = "H" - Battery offline status
"0x00000008"	Offline
"0x00000020"	The cover is open
"0x00000040"	Paper is being fed by a paper feed switch operation
"0x00000100"	Waiting to be brought back online
"0x00000200"	The paper feed switch is being pressed (ON)
"0x00000400"	A mechanical error occurred
"0x00000800"	An autocutter error occurred
"0x00002000"	An unrecoverable error occurred
"0x00004000"	An automatically recoverable error occurred
"0x00020000"	No paper in roll paper near end sensor
"0x00080000"	No paper in roll paper end sensor
"0x01000000"	A buzzer is on (only for applicable devices)
"0x80000000" *	The spooler has stopped

* ePOS-Device Service Ver. 2.6 or later versions supported

□ battery

- Depending on the battery situation, an OR function is acquired in a decimal expression from the following values.

Status of power

Attribute value	Description
0x30XX	The AC adapter is connected
0x31XX	The AC adapter is not connected

Remaining battery

Attribute value	Description
0xXX36	Battery amount 6
0xXX35	Battery amount 5
0xXX34	Battery amount 4
0xXX33	Battery amount 3
0xXX32	Battery amount 2
0xXX31	Battery amount 1 (Near end)
0xXX30	Battery amount 0 (Real end)



0 is indicated when the model doesn't have a battery installed.

Example

```
<!-- Printer status -->
<!-- Successfully performed. -->
  <response success="true" code="" status="" />
<!-- When the printer ran out of paper during printing and failed to print. -->
<response xmlns="http://www.epson-pos.com/schemas/2011/03/epos-print"
  success="false" code="EPTR_REC_EMPTY" status="252641308" />
```

<text>

Prints a character string. Also, configure the string-related settings such as style, print position, and line feed space.



After printing text, to print content other than text content, execute line feed or paper feed.



In page mode, characters are laid out in the current print position with the reference point being the character baseline dot ([Printer specifications \(p.191\)](#)).

Entity Reference

To write symbols required for printer control such as horizontal tab, line feed, and the following symbols, use their corresponding entity references shown as follows:

Function (symbol)	Entity Reference
&	&
'	'
>	>
<	<
"	"
Horizontal tab(HT)		 or

Line feed (LF)	
 or

Attributes Available for Each Printing Method

Attribute	Description	Receipt/ Label printing	Slip printing	Endorse printing	Endorse printing 40cpl mode	Page
lang	Language	●	●	▲	▲	92
font	Character font	●	●	●	▲	93
smooth	smoothing	●	▲	▲	▲	93
dw	Double-sized width	●	●	●	▲	93
dh	Double-sized height	●	●	●	▲	93
width	Horizontal scale	●	●	●	▲	93
height	Vertical scale	●	●	●	▲	94
reverse	Inversion of black and white parts of characters	●	▲	▲	▲	94
ul	underline	●	●	●	▲	94
em	Emphasized printing	●	●	●	▲	94
color	Character color	●	▲	▲	▲	94
x	Print start position	●	●	●	●	94
y	Vertical print start position	●	●	▲	▲	94
align	Alignment	●	●	●	▲	95
rotate	Rotated printing	●	●	●	●	95
linespc	Per-line paper feed amount	●	●	●	▲	95

Attribute

- lang : Specifies the target language.

Attribute value	Description
en (default)	English (ANK)
de	German (ANK)
fr	French (ANK)
it	Italian (ANK)
es	Spanish (ANK)
ja	Japanese (International character set changes to Japan.)
ja-jp	Japanese (International character set changes to Japan.)
ko	Korean (International character set changes to Korean.)
ko-kr	Korean (International character set changes to Korean.)
zh-hans	Simplified Chinese * (International character set changes to China.)
zh-cn	Simplified Chinese (International character set changes to China.)
zh-hant	Traditional Chinese *
zh-tw	Traditional Chinese
th	Thai (Only for South Asia model.)
Language code besides above	English (ANK)

* ePOS-Device Service Ver. 2.2 or later versions supported

	Characters not installed in a printer cannot be printed.
	For printable character code, refer to the Technical Reference Guide of your printer.

Depending on language specification, a part of characters is printed as follows.

Language	Characters \$(U+0024)	Characters \ (U+005C)
Japanese	\$	¥
Korean	\$	₩
Simplified Chinese	¥	\
Traditional Chinese	\$	\

- font : Specifies the character font.

Attribute value	Description
"font_a" (default)	Font A
"font_b"	Font B
"font_c"	Font C
"font_d"	Font D
"font_e"	Font E
"special_a"	Special font A
"special_b"	Special font B



When Japanese is specified, characters in Shift-JIS can be printed.

- smooth : Sets smoothing. When smoothing is enabled, the print quality of characters will be improved.

Attribute value	Description
"true" or "1"	Specifies smoothing.
"false" or "0" (default)	Cancels smoothing

- dw : Sets the double-sized width.

Attribute value	Description
"true" or "1"	Specifies the double-sized width.
"false" or "0" (default)	Cancels the double-sized width



When the double-sized width attribute and the width attribute are specified in one element at the same time, priority is given to the scale specification for the width attribute.

- dh : Sets the double-sized height.

Attribute value	Description
"true" or "1"	Specifies the double-sized height
"false" or "0" (default)	Cancels the double-sized height



When the double-sized height attribute and the height attribute are specified in one element at the same time, priority is given to the scale specification for the height attribute.

- width : Specifies the horizontal scale.

Attribute value	Description
Integer from "1" to "8" (default : "1")	Horizontal scale



When the double-sized width attribute and the width attribute are specified in one element at the same time, priority is given to the scale specification for the width attribute.

- height : Specifies the vertical scale.

Attribute value	Description
Integer from "1" to "8" (default : "1")	Vertical scale



When the double-sized height attribute and the height attribute are specified in one element at the same time, priority is given to the scale specification for the height attribute.

- reverse : Sets the inversion of black and white parts of characters.

Attribute value	Description
"true" or "1"	Specifies the inversion of black and white parts of characters.
"false" or "0" (default)	Cancels the inversion of black and white parts of characters.

- ul : Sets underlining.

Attribute value	Description
"true" or "1"	Specifies underlining.
"false" or "0" (default)	Cancels underlining.

- em : Sets emphasized printing.

Attribute value	Description
"true" or "1"	Specifies emphasized printing.
"false" or "0" (default)	Cancels emphasized printing.

- color : Specifies the character color.

Attribute value	Description
"none"	Characters are not printed.
"color_1" (default)	First color
"color_2"	Second color
"color_3"	Third color
"color_4"	Fourth color

- x : (Default : "0")
Specifies the print start position of the text in dots.
In page mode, this specification is the same as the specification for the horizontal print position by the position element, but the print position in the horizontal direction only will be moved.
- y : (Default : 21)
Specifies the vertical print start position of the text in dots.
(in ePOS-Device Service Ver.2.2 and later)
In page mode, the specification is the same as the specification for the vertical print position by the position element, but the print position in the vertical direction only will be moved.



Use this attribute inside the page element.

- align : Specifies the alignment.

Attribute value	Description
"left" (default)	Alignment to the left
"center"	Alignment to the center
"right"	Alignment to the right



- For the standard mode, specify the align attribute "when at the start of a line".
- In page mode, even if align is specified, it is disabled.
- The align attribute set in this element also applies to the align attribute in each of the image, logo, barcode, and symbol elements.

- rotate : Sets rotated printing of text.

Attribute value	Description
"true" or "1"	Specifies rotated printing of text.
"false" or "0" (default)	Cancels rotated printing of text.



- For the standard mode, specify the rotate attribute "when at the start of a line".
- In page mode, even if rotate is specified, it is disabled.
For the page mode, set the print direction as right to left (right_to_left) using the direction element to perform 180-degree rotated printing of text.
- The rotate attribute set in this element also applies to the rotate attribute in each of the barcode and symbol elements.

- linespc : (Default : Differ according to each model. Refer to "[Printer specifications](#)".)
Specifies the per-line paper feed amount in dots.

Example

To print a character string based on the following settings:

Item	Setting
Language	English
Smoothing	Enable
Align	Center Align
Font	Font A
Double-sizing	Double width and height characters
Underline	Enable

```
<text lang="en" />
<text smooth="true" />
<text align="center" />
<text font="font_a" />
<text dw="true" dh="true" />
<text ul="true" />
<text>Hello, World!&#10;</text>
```

<feed>

Feeds paper. Specifies the paper feed amount in dots or in lines. When no paper feed amount is specified, performs single-line paper feed (line feed). In addition, the per-line line feed space can be set.

Attributes Available for Each Printing Method

Attribute	Description	Receipt/ Label printing	Slip printing	Endorse printing	Endorse printing 40cpl mode	Page
unit	Paper feed amount in dots	●	●	●	●	96
line	Paper feed amount in lines	●	●	●	▲	96
linespc	Per-line paper feed amount	●	●	●	●	96
pos	Paper feed position of label paper/black mark paper	●	▲	▲	▲	96

Attribute

- unit : Specifies the paper feed amount in dots.
- line : Specifies the paper feed amount in lines.
- linespc : (Default : Differ according to each model. Refer to "[Printer specifications](#)".)
Specifies the per-line paper feed amount in dots.
- pos : Specifies paper feed position of label paper/black mark paper

Attribute value	Description
"peeling"	Feeds to the peeling position.
"cutting"	Feeds to the cutting position.
"current_tof"	Feeds to the top of the current label.
"next_tof"	Feeds to the top of the next label.



- When the per-line paper feed amount is 30 dots, up to 240 lines can be specified.
- The standard mode retains the line feed space separate from the line feed space retained by the page mode. When the linespc attribute is specified in this element, it will affect the linespc attribute in each of the <text> and <feed> elements that follow.
- In the page mode, pos attribute cannot be specified.

Example

```
<!-- Feed paper 3 lines forward -->  
<feed line="3" />
```

```
<!-- Start a new line after printing a character string -->  
<text>Hello</text><feed />  
<text>World</text><feed />
```

```
<!-- To print while peeling the label one by one -->  
<feed pos="current_tof" />  
<barcode type="code39" hri="below">0001</barcode>  
<feed pos="peeling" />
```

<image>

Print a raster graphic bit-image. (Data type xs:base64Binary)

Raster graphics refer to data generated by horizontally scanning the pixels of an image from the top left corner of that image as the origin point.

Data is generated based on 1 bit per pixel for two-tone images, and 4 bits per pixel for 16-tone images, starting with the upper bits of byte data. And zero-padding is performed so that scan data per line can be treated on a byte basis.



- To print a raster image at high speed, specify "left" for the align attribute and specify a multiple of 8 that does not exceed the printer paper width for the width attribute value.
- Page mode does not support printing in multiple tones. Multiple tone graphics can only be printed in standard mode.



- In page mode, a raster image is laid out in the current print position with the reference point being its bottom left dot. The print position will not move.
- In page mode, set the print position so that a raster image does not extend beyond the print area.
- When multiple tones are set for raster images, intermitting printing may occur because the amount of data to print increases and white stripes may appear in the print result.
- The scan quality of barcodes/2D-codes printed as multiple-tone raster images cannot be guaranteed. Print them as two-tone images.
- To create raster graphic bit-images, use the ePOS-Print XML generator tool or your own application. Note the following when creating raster graphic images using your own application.
Two-tone image: Specify a multiple of 8 for the image width or fill the missing bits with zeros.
16-tone image: Specify a multiple of 2 for the image width or fill the missing bits with zeros.

Attribute

- width : (Required attribute)
Specifies the image width in dots.
- height : (Required attribute)
Specifies the image height in dots.
- color : (Optional attribute)
Specifies the character color.

Attribute value	Description
"none"	Characters are not printed.
"color_1" (default)	First color
"color_2"	Second color
"color_3"	Third color
"color_4"	Fourth color

- align : Specifies the print position.

Attribute value	Description
"left" (default)	Alignment to the left
"center"	Alignment to the center
"right"	Alignment to the right



- In page mode, even if align is specified, it is disabled.
- For the standard mode, specify the align attribute "when at the start of a line".
- The align attribute set in this element also applies to the align attribute in each of the text, logo, barcode, and symbol elements.

- mode : (Optional attribute)
Specifies the color mode.

Attribute value	Description
"mono" (default)	Monochrome (two-tone)
"gray16"	Multiple tones (16-tone)

Example

```
<!-- Print an 8 dot wide, 8 dot high filled-in raster image -->
<image width="8" height="8">/////////8=</image>
```

<logo>

Prints a logo registered in the NV memory of the printer.



Page mode does not support printing in multiple tones. Multiple tone graphics can only be printed in standard mode.



- Using model-dedicated utility or logo registration utility (TMFLogo), register a logo in the printer in advance.
- In page mode, a logo is laid out in the current print position with the reference point being its bottom left dot.

Attribute

- **key1 :** (Required attribute)
Specifies the value of the key code 1 set at the time of NV logo registration.
- **key2 :** (Required attribute)
Specifies the value of the key code 2 set at the time of NV logo registration. Be sure to specify this value.
- **align :** Specifies the print position.

Attribute value	Description
"left" (default)	Alignment to the left
"center"	Alignment to the center
"right"	Alignment to the right



- In page mode, even if align is specified, it is disabled.
- For the standard mode, specify the align attribute "when at the start of a line".
- The align attribute set in this element also applies to the align attribute in each of the text, image, barcode, and symbol elements.

Example

```
<!-- Print a NV logo with key1 registered as 48 and key2 registered as 48 -->  
<logo key1="48" key2="48" />
```

<barcode>

Prints a barcode.

Specifies the barcode data using a character string. (Data type xs:string)



- A barcode will not be printed if settings not compliant with the barcode standard are made, or if the barcode becomes larger than the printer's print area. And an error response will not be returned, either.
- In page mode, a barcode is laid out in the current print position with the reference point being its bottom left dot (except for HRI).

Barcode Type

The following barcodes can be specified:

Barcode type	Barcode data specification procedure
UPC-A	When an 11-digit number is specified, a check digit is automatically added. When a 12-digit number is specified, the 12th digit is processed as a check digit but the check digit is not validated.
UPC-E	Specify 0 as the first digit. Specify the manufacturer code in the digits 2 to 6. Specify (right-align) the item code in the digits 7 to 11. The number of item code digits varies depending on the manufacturer code. Specify 0s in empty digits. When an 11-digit number is specified, a check digit is automatically added. When a 12-digit number is specified, the 12th digit is processed as a check digit but the check digit is not validated.
EAN13	When an 12-digit number is specified, a check digit is automatically added. When a 13-digit number is specified, the 12th digit is processed as a check digit but the check digit is not validated.
JAN13	
EAN8	When a 7-digit number is specified, a check digit is automatically added. When an 8-digit number is specified, the 8th digit is processed as a check digit but the check digit is not validated.
JAN8	
CODE39	When the first character is *, the character is processed as the start character. In other cases, a start character is automatically added.
ITF	Start and stop codes are automatically added. Check digits are not added or validated.
CODABAR	Specify a start character (A to D, a to d). Specify a stop character (A to D, a to d). Check digits are not added or validated.
CODE93	Start and stop characters are automatically added. A check digit is automatically calculated and added.

Barcode type	Barcode data specification procedure																		
CODE128	Specify a start character (CODE A, CODE B, CODE C). A stop character is automatically added. A check digit is automatically calculated and added. To encode each of the following characters, specify two characters starting with the character "{": <table> <tr><td>FNC1:</td><td>{1</td></tr> <tr><td>FNC2:</td><td>{2</td></tr> <tr><td>FNC3:</td><td>{3</td></tr> <tr><td>FNC4:</td><td>{4</td></tr> <tr><td>CODE A:</td><td>{A</td></tr> <tr><td>CODE B:</td><td>{B</td></tr> <tr><td>CODE C:</td><td>{C</td></tr> <tr><td>SHIFT:</td><td>{S</td></tr> <tr><td>{:</td><td>{{</td></tr> </table>	FNC1:	{1	FNC2:	{2	FNC3:	{3	FNC4:	{4	CODE A:	{A	CODE B:	{B	CODE C:	{C	SHIFT:	{S	{:	{{
FNC1:	{1																		
FNC2:	{2																		
FNC3:	{3																		
FNC4:	{4																		
CODE A:	{A																		
CODE B:	{B																		
CODE C:	{C																		
SHIFT:	{S																		
{:	{{																		
GS1-128	A start character, FNC1, a check digit, and a stop character are automatically added. To automatically calculate and add a check digit for an application identifier (AI) and the subsequent data, specify the character "*" in the position of the check digit. You can enclose an application identifier (AI) in parentheses. The parentheses are used as HRI print characters and are not encoded as data. You can insert spaces between an application identifier (AI) and data. The spaces are used as HRI print characters and are not encoded as data. To encode each of the following characters, specify two characters starting with the character "{": <table> <tr><td>FNC1:</td><td>{1</td></tr> <tr><td>FNC3:</td><td>{3</td></tr> <tr><td>(:</td><td>{{</td></tr> <tr><td>):</td><td>{}</td></tr> <tr><td>*:</td><td>{*</td></tr> <tr><td>{:</td><td>{{</td></tr> </table>	FNC1:	{1	FNC3:	{3	(:	{{	):	{}	*:	{*	{:	{{						
FNC1:	{1																		
FNC3:	{3																		
(:	{{																		
):	{}																		
:	{																		
{:	{{																		
GS1 DataBar Omnidirectional	Specify a 13-digit global trade item number (GTIN) not including an application identifier (AI) or a check digit.																		
GS1 DataBar Truncated																			
GS1 DataBar Limited																			
GS1 DataBar Expanded	You can enclose an application identifier (AI) in parentheses. The parentheses are used as HRI print characters and are not encoded as data. To encode each of the following characters, specify two characters starting with the character "{": <table> <tr><td>FNC1:</td><td>{1</td></tr> <tr><td>(:</td><td>{{</td></tr> <tr><td>):</td><td>{}</td></tr> </table>	FNC1:	{1	(:	{{	):	{}												
FNC1:	{1																		
(:	{{																		
):	{}																		

To specify binary data that cannot be represented by character strings, use the following escape sequences.

String	Description
\xnn	Control code
\\	Back slash

Attribute

- type : (Required attribute)
Specifies the barcode type.

Attribute value	Description
"upc_a"	UPC-A
"upc_e"	UPC-E
"ean13"	EAN13
"jan13"	JAN13
"ean8"	EAN8
"jan8"	JAN8
"code39"	CODE39
"itf"	ITF
"codabar"	CODABAR
"code93"	CODE93
"code128"	CODE128
"gs1_128"	GS1-128
"gs1_databar_omnidirectional"	GS1 DataBar Omnidirectional
"gs1_databar_truncated"	GS1 DataBar Truncated
"gs1_databar_limited"	GS1 DataBar Limited
"gs1_databar_expanded"	GS1 DataBar Expanded

- hri : Specifies the HRI position.

Attribute value	Description
"none" (default)	HRI not printed
"above"	Above the barcode
"below"	Below the barcode
"both"	Both above and below the barcode

- font : Specifies the HRI font.

Attribute value	Description
"font_a" (default)	Font A
"font_b"	Font B
"font_c"	Font C
"font_d"	Font D
"font_e"	Font E

- width : (Default "3")
Specifies the width of each module in dots. Specifies an integer from 2 to 6.
- height : (Default "162")
Specifies the barcode height in dots.

- align : Specifies the print position.

Attribute value	Description
"left" (default)	Alignment to the left
"center"	Alignment to the center
"right"	Alignment to the right



The align attribute set in this element also applies to the align attribute in each of the text, image, logo, and symbol elements.

- rotate : (data type xs:boolean)
Specifies the rotate printing.

Attribute value	Description
"true" or "1"	Specifies rotated printing of text.
"false" or "0" (default)	Cancels rotated printing of text.



The rotate attribute set in this element also applies to the rotate attribute in each of the text and symbol elements.

Example

```
<!-- Print barcodes -->
<barcode type="upc_a" width="2" height="64" hri="below">01234567890</barcode>
<barcode type="upc_e">01234500005</barcode>
<barcode type="ean13">201234567890</barcode>
<barcode type="jan13">201234567890</barcode>
<barcode type="ean8">2012345</barcode>
<barcode type="jan8">2012345</barcode>
<barcode type="code39">ABCDE</barcode>
<barcode type="itf">012345</barcode>
<barcode type="codabar">A012345A</barcode>
<barcode type="code93">ABCDE</barcode>
<barcode type="code128">{Babcde</barcode>
<barcode type="gs1_128">(01)201234567890*</barcode>
<barcode type="gs1_databar_omnidirectional">0201234567890</barcode>
<barcode type="gs1_databar_truncated">0201234567890</barcode>
<barcode type="gs1_databar_limited">0201234567890</barcode>
<barcode type="gs1_databar_expanded">(01)2012345678903</barcode>
```

<symbol>

Prints a 2D-code.

Specifies the 2D-code using a character string. (Data type xs:string)



- A 2D-code will not be printed if settings not compliant with the 2D-code standard are made, or if the 2D-code becomes larger than the printer's print area. And an error response will not be returned, either.
- In standard mode, a more than 831 dot high 2D-code cannot be printed.
- In page mode, a 2D-code is laid out in the current print position with the reference point being its top left dot.

2D-Code Type

The following 2D-codes can be specified:

Type	Description
PDF417	Convert the character string to the string in UTF-8, apply the escape sequence, and then encode the string. The data area can contain up to 928 code words in a maximum of 90 rows, each of which can contain up to 30 code words.
QR Code	Convert the character string to the string in Shift-JIS, apply the escape sequence, and then encode the string based on the data type as shown below. Number: 0 to 9 Alphanumeric character: 0 to 9, A to Z, space, \$, %, *, +, -, ., /, : Kanji character: Shift-JIS string 8-bit, byte data: 0x00 to 0xff
Micro QR Code	

Type	Description
MaxiCode	Convert the character string to the string in UTF-8, apply the escape sequence, and then encode the string. In Modes 2 and 3, when the first piece of data is 0>\ x1e01\x1dy (where yy is a two-digit number), this is processed as the message header, and the subsequent data is processed as the primary message. In other cases, from the first piece of data, data is processed as the primary message. Specify the primary message in the following format: • In Mode 2 Postal code (1- to 9-digit number) GS:(\x1d) ISO country code (1- to 3-digit number) GS:(\x1d) Service class code (1- to 3-digit number) • In Mode 3 Postal code (1 to 6 pieces of data convertible by Code Set A) GS:(\x1d) ISO country code (1- to 3-digit number) GS:(\x1d) Service class code (1- to 3-digit number)
GS1 DataBar Stacked	Convert the character string to the string in UTF-8, apply the escape sequence, and then encode the string. Specify a 13-digit global trade item number (GTIN) not including an application identifier (AI) or a check digit.
GS1 DataBar Stacked Omnidirectional	
GS1 DataBar Expanded Stacked	Convert the character string to the string in UTF-8, apply the escape sequence, and then encode the string. You can enclose an application identifier (AI) in parentheses. The parentheses are used as HRI print characters and are not encoded as data. To encode each of the following characters, specify two characters starting with the character "{": FNC1: {1 (: {(): {)
Aztec Code	After converting the character string to UTF-8, conduct the escape sequence and encode.
DataMatrix	After converting the character string to UTF-8, conduct the escape sequence and encode.

To specify binary data that cannot be represented by character strings, use the following escape sequences.

String	Description
\xnn	Control code
\\	Back slash

Attribute

- type : (Required attribute)
Specifies the 2D-code type.

Attribute value	Type
"pdf417_standard"	Standard PDF417
"pdf417_truncated"	Truncated PDF417
"qrcode_model_1"	QR Code Model 1
"qrcode_model_2"	QR Code Model 2
"qrcode_micro" *2	Micro QR Code
"maxicode_mode_2"	MaxiCode Mode 2
"maxicode_mode_3"	MaxiCode Mode 3
"maxicode_mode_4"	MaxiCode Mode 4
"maxicode_mode_5"	MaxiCode Mode 5
"maxicode_mode_6"	MaxiCode Mode 6
"gs1_databar_stacked"	GS1 DataBar Stacked
"gs1_databar_stacked_omnidirectional"	GS1 DataBar Stacked Omnidirectional
"gs1_databar_expanded_stacked"	GS1 DataBar Expanded Stacked
"azteccode_fullrange" *1	Aztec Code Full-Range Mode
"azteccode_compact" *1	Aztec Code Compact Mode
"datamatrix_square" *1	DataMatrix ECC200 square
"datamatrix_rectangle_8" *1	DataMatrix ECC200 rectangle, 8 lines
"datamatrix_rectangle_12" *1	DataMatrix ECC200 rectangle, 12 lines
"datamatrix_rectangle_16" *1	DataMatrix ECC200 rectangle, 16 lines

*1 ePOS-Device Service Ver. 2.2 or later versions supported

*2 ePOS-Device Service Ver. 2.6 or later versions supported

- **level :** Specifies the error correction level.

Attribute value	Description
"level_0"	PDF417 error correction level 0
"level_1" (default)	PDF417 error correction level 1
"level_2"	PDF417 error correction level 2
"level_3"	PDF417 error correction level 3
"level_4"	PDF417 error correction level 4
"level_5"	PDF417 error correction level 5
"level_6"	PDF417 error correction level 6
"level_7"	PDF417 error correction level 7
"level_8"	PDF417 error correction level 8
"level_l"	QR Code error correction level L
"level_m"	QR Code error correction level M
"level_q"	QR Code error correction level Q
"level_h"	QR Code error correction level H
Integer from "5" to "95" *	Aztec Code error correction level (Default : 23)
"default"	Default level

* ePOS-Device Service Ver. 2.2 or later versions supported



- Select the level according to the 2D-code type.
- Select the default for MaxiCode and two-dimensional GS1 DataBar.
- Micro QR Code does not support level_h.

- **width :** Specifies the width of each module in dots.

2D-Code Type	Valid value range	Default
PDF417	2 to 8	width="3"
QR Code	3 to 16	width="3"
MaxiCode	Ignored	
2-dimensional GS1 DataBar	2 to 8	width="2"
Aztec Code	2 to 16	width="3"
DataMatrix	2 to 16	width="3"

- **height :** Specifies the height of each module in dots.

2D-Code Type	Valid value range	Default
PDF417	2 to 8 (Magnification for width)	height="3"
QR Code	Ignored	
MaxiCode		
GS1 DataBar		
Aztec Code		
DataMatrix		

- size :

2D-Code	Default	Description
PDF417	size="0"	Specifies the number of code words for each row
QR Code	Ignored	
MaxiCode		
2-dimentional GS1 DataBar (Expanded Stacked)	size="0" (auto)	Specifies the maximum width for the barcode (106 or above)
(Others)	Ignored	
Aztec Code	Ignored	
DataMatrix		

- align : Specifies the print position.

Attribute value	Description
"left" (default)	Alignment to the left
"center"	Alignment to the center
"right"	Alignment to the right



The align attribute set in this element also applies to the align attribute in each of the text, image, logo, and barcode elements.

- rotate : Specifies the rotate printing.

Attribute value	Description
"true" or "1"	Specifies rotated printing of text.
"false" or "0" (default)	Cancels rotated printing of text.



- For the standard mode, specify the rotate attribute "when at the start of a line".
- In page mode, even if rotate is specified, it is disabled.
- In the page mode, set the printing direction to "right_to_left" with "direction" and conduct 180 rotation printing.
- The rotate attribute set in this element also applies to the rotate attribute in each of the text and barcode elements.

Example

```
<!-- Print 2D-codes -->
<symbol type="pdf417_standard">ABCDE</symbol>
<symbol type="qrcode_model_2" level="level_q">ABCDE</symbol>
<symbol type="maxicode_mode_2">908063840\x1d850\x1d001\x1d\x04</symbol>
<symbol type="gs1_databar_stacked">0201234567890</symbol>
<symbol type="gs1_databar_stacked_omnidirectional">0201234567890</symbol>
<symbol type="gs1_databar_expanded_stacked">(01)02012345678903</symbol>
```

<hline>

Draws horizontal lines.



Not available in page mode.

Attribute

- **x1 :** (Required attribute)
Specifies the draw start position of the horizontal line (in dots).
- **x2 :** (Required attribute)
Specifies the draw end position of the horizontal line (in dots).
- **style :** (Optional attribute)
Specifies the line type.

Attribute value	Description
"thin"(default)	Solid line: Thin
"medium"	Solid line: Medium
"thick"	Solid line: Thick
"thin_double"	Double line: Thin
"medium_double"	Double line: Medium
"thick_double"	Double line: Thick

Example

```
<!-- Draw double lines(From the left end: 100th dot to 200th dot) -->
<!--           (From the left end: 400th dot to 500th dot) -->
<hline x1="100" x2="200" style="thin_double" />
<hline x1="400" x2="500" style="thin_double" />
```

<vline-begin>

Starts to draw vertical lines.



Not available in page mode.



A vertical line is drawn up to the end position specified by the vline-end element ([p. 112](#)).
Use this element together with the vline-end element.

Attribute

- **x :** (Required attribute)
Specifies the start position of the horizontal line (in dots).
- **style :** (Optional attribute)
Specifies the line type.

Attribute value	Description
"thin"(default)	Solid line: Thin
"medium"	Solid line: Medium
"thick"	Solid line: Thick
"thin_double"	Double line: Thin
"medium_double"	Double line: Medium
"thick_double"	Double line: Thick

Example

```
<!-- Draw thin lines from 100th to 200th dot from the left end -->
<vline-begin x="100" />
<vline-begin x="200" />
<feed unit="100" />
<vline-end x="100" />
<vline-end x="200" />
```

<vline-end>

Ends to draw vertical lines.



Not available in page mode.



Use this element together with the vline-begin element ([p. 111](#)).

Attribute

- **x :** (Required attribute)
Specifies the end position of the horizontal line (in dots).
- **style :** (Optional attribute)
Specifies the line type.

Attribute value	Description
"thin"(default)	Solid line: Thin
"medium"	Solid line: Medium
"thick"	Solid line: Thick
"thin_double"	Double line: Thin
"medium_double"	Double line: Medium
"thick_double"	Double line: Thick

Example

```
<!-- Draw thin lines from 100th to 200th dot from the left end -->  
  <vline-begin x="100" />  
  <vline-begin x="200" />  
  <feed unit="100" />  
  <vline-end x="100" />  
  <vline-end x="200" />
```

<page>

Changes the mode from standard mode to page mode.

Elements via <page>

Via <page>, the following elements can be used.

Element	Description
<text>	Prints text
<feed>	Paper feed
<image>	Prints a raster image
<logo>	Prints a NV logo
<barcode>	Prints a barcode
<symbol>	Prints a 2D-code
<area>	Sets a print area
<direction>	Sets a print direction
<position>	Sets a print position
<line>	Draws a line
<rectangle>	Draws a rectangle
<command>	Inserts a command

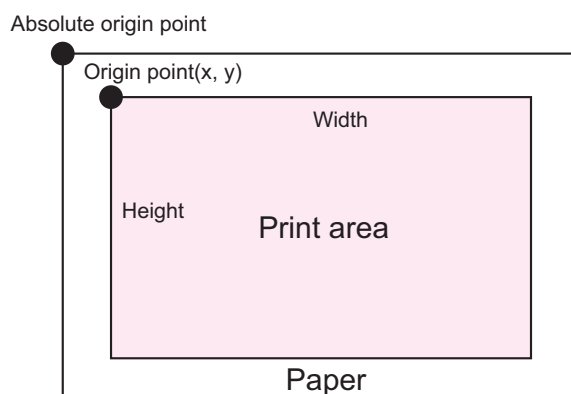
Example

```
<!-- Print "ABCDE" in page mode -->
  <page>
    <text>ABCDE</text>
  </page>
```

<area>

Specifies the page mode print area.

Specifies the origin point, width, and height based on the absolute origin point to set the print area. The absolute origin point is a dot diagonally above the top left corner of the printable area.



Specify the print area according to the print content. If print data extends beyond the print area, the print result is such that the whole data cannot be printed completely.



Use this element inside the page element.

Attribute

- **x :** (Required attribute, Default "0")
Specifies the origin of the horizontal axis (in dots).
- **y :** (Required attribute, Default "0")
Specifies the origin of the vertical axis (in dots).
- **width :** (Required attribute,
Default :Differ according to each model. Refer to ["Printer specifications".](#))
Specifies the width of the print area (in dots).
- **height :** (Required attribute,
Default :Differ according to each model. Refer to ["Printer specifications".](#))
Specifies the height of the print area (in dots).



Determine the width and height of the print area according to the print direction setting. Otherwise, the print data might not be printed completely.

Example

```
<!-- Print the characters "ABCDE" by specifying the print area as origin (100, 50), width: 200  
dots, and height: 30 dots -->  
<page>  
  <area x="100" y="50" width="200" height="30" />  
  <text>ABCDE</text>  
</page>
```

<direction>

Specifies the page mode print direction.

Specifies the print direction and rotates the print area. Following the rotation of the print area, the start point of the print area moves.



Use this element inside the page element.

Attribute

- dir : (Required attribute)
Specifies the rotation direction.

Attribute value	Description
"left_to_right" (default)	Left to right (No rotation. Data is printed from the top left corner to the right.)
"bottom_to_top"	Bottom to top (Counterclockwise rotation by 90 degrees. Data is printed from the bottom left corner to the top.)
"right_to_left"	Right to left (Rotation by 180 degrees. Data is printed from the bottom right corner to the left.)
"top_to_bottom"	Top to bottom (Clockwise rotation by 90 degrees. Data is printed from the top right corner to the bottom.)

Example

```
<!-- Print the characters "ABCDE" by rotating them 90 degrees clockwise -->
<page>
  <direction dir="top_to_bottom" />
  <text>ABCDE</text>
</page>
```

<position>

Specifies the page mode print position.

Specifies the print position based on the start point of the print area. The start point of the print area moves by following the rotation of the print area.



Use this element inside the page element.

Attribute

- **x :** (Required attribute, Default "0")
Specifies the origin of the horizontal axis (in dots).
- **y :** (Required attribute, Default "21")
Specifies the origin of the vertical axis (in dots).



Specify the print start position (coordinates) according to the content to be printed. Refer to the following.

- * To print a character string:
Specify the left end of the baseline for the first character. This can be omitted for left-aligned printing of standard-sized characters. To print double-sized height characters, specify a value equal to or greater than "42" for y.
- * To print a barcode:
Specify the bottom left of the symbol. And specify the barcode height for y.
- * To print a graphic/logo:
Specify the bottom left of the graphic data. And specify the graphic data height for y.
- * To print a 2D-code:
Specify the top left of the symbol. This can be omitted when printing from the top left.

Example

```
<!-- Specify (50,30) for the print start position in the area specified by the addPageArea  
method and print the characters "ABCDE" -->
```

```
<page>  
  <area x="100" y="50" width="200" height="100" />  
  <position x="50" y="30" />  
  <text>ABCDE</text>  
</page>
```

<line>

Draws a line in page mode.



Diagonal lines cannot be drawn.



Use this element inside the page element.

Attribute

- **x1 :** (Required attribute)
Specifies the drawing start position of the horizontal line (in dots).
- **y1 :** (Required attribute)
Specifies the drawing start position of the vertical line (in dots).
- **x2 :** (Required attribute)
Specifies the drawing end position of the horizontal line (in dots).
- **y2 :** (Required attribute)
Specifies the drawing end position of the vertical line (in dots).
- **style :** (Optional attribute)
Specifies the line type.

Attribute value	Description
"thin" (default)	Solid line: Thin
"medium"	Solid line: Medium
"thick"	Solid line: Thick
"thin_double"	Double line: Thin
"medium_double"	Double line: Medium
"thick_double"	Double line: Thick

Example

```
<!-- Draw a thin solid line between the start position (100, 0) and the end position (500, 0) -
->
<page>
  <line x1="100" y1="0" x2="500" y2="0" style="thin" />
</page>
```

<rectangle>

Draws a rectangle in page mode.



- Use this element inside the page element.
- Not available in standard mode. Use the hline, vline-begin, and vline-end elements instead.

Attribute

- x1 : (Required attribute)
Specifies the drawing start position of the horizontal line (in dots).
- y1 : (Required attribute)
Specifies the drawing start position of the vertical line (in dots).
- x2 : (Required attribute)
Specifies the drawing end position of the horizontal line (in dots).
- y2 : (Required attribute)
Specifies the drawing end position of the vertical line (in dots).
- style : (Optional attribute)
Specifies the line type.

Attribute value	Description
"thin" (default)	Solid line: Thin
"medium"	Solid line: Medium
"thick"	Solid line: Thick
"thin_double"	Double line: Thin
"medium_double"	Double line: Medium
"thick_double"	Double line: Thick

Example

```
<!-- Draw a rectangle with a thin double line, with the start position (100, 0) and the end
position (500, 200) as its vertexes -->
<page>
  <rectangle x1="100" y1="0" x2="500" y2="200" style="thin-double" />
</page>
```

<cut>

Sets paper cut.

	Available in standard mode. Not available in page mode.
	Execute "when at the top of a line". When the paper is cut, the state turns to "when at the top of a line".

Attribute

- type : (Optional attribute)
Specifies the paper cut type.

Attribute value	Description
"no_feed"	Cut without feeding (The paper is cut without being fed.)
"feed" (default)	Feed cut (The paper is fed to the cut position and then is cut.)
"reserve"	Cut reservation (Printing continues until the cut position is reached, at which the paper is cut.)

Example

```
<!-- Perform feed cut operation -->
<cut type="feed" />
```

<pulse>

Outputs signals to the drawer kick connector.
Depending on the model, a buzzer can be sounded.



- Not available in page mode.
- The drawer and the buzzer cannot be used together.

Attribute

- drawer : (Optional attribute)
Specifies the drawer kick connector.

Attribute value	Description
"drawer_1" (default)	Pin 2 of the drawer kick-out connector
"drawer_2"	Pin 5 of the drawer kick-out connector

- time : (Optional attribute)
Specifies the ON time of the drawer kick signal.

Attribute value	Description
"pulse_100" (default)	100 ms
"pulse_200"	200 ms
"pulse_300"	300 ms
"pulse_400"	400 ms
"pulse_500"	500 ms

Example

```
<!-- Send a 100msec pulse signal to the pin 2 of the drawer kick connector -->  
<pulse drawer="drawer_1" time="pulse_100" />
```

<sound>

Sounds a buzzer.



- Not available in page mode.
- The buzzer function and the drawer cannot be used together.
- This element is not available if the printer is not provided with the buzzer.

Attribute

- pattern : (Optional attribut)
Specifies the buzzer pattern.

Attribute value	Description
"none"	Stop
"pattern_a" (default)	Pattern A (Optional Buzzer)
"pattern_b"	Pattern B (Optional Buzzer)
"pattern_c"	Pattern C (Optional Buzzer)
"pattern_d"	Pattern D (Optional Buzzer)
"pattern_e"	Pattern E (Optional Buzzer)
"error"	Error sound pattern (Optional Buzzer)
"paper_end"	Pattern when there is no paper (Optional Buzzer)
"pattern_1" *	Pattern 1 (Internal Buzzer)
"pattern_2" *	Pattern 2 (Internal Buzzer)
"pattern_3" *	Pattern 3 (Internal Buzzer)
"pattern_4" *	Pattern 4 (Internal Buzzer)
"pattern_5" *	Pattern 5 (Internal Buzzer)
"pattern_6" *	Pattern 6 (Internal Buzzer)
"pattern_7" *	Pattern 7 (Internal Buzzer)
"pattern_8" *	Pattern 8 (Internal Buzzer)
"pattern_9" *	Pattern 9 (Internal Buzzer)
"pattern_10" *	Pattern 10 (Internal Buzzer)

* ePOS-Device Service Ver. 2.2 or later versions supported

- repeat : (Optional attribut, Default: "1")
Specifies the number of repeats.

Attribute value	Description
"0"	The buzzer does not stop.
"1" to "255"	Number of repeats



To stop the buzzer after designating "0" for "repeat", designate "none" for this element's "pattern", and make the request once more.

- cycle : (in ePOS-Device Service Ver.2.2 and later)
Specifies the buzzer sounding cycle (in units of milliseconds)



"cycle" is enabled by any of "pattern_1" to "pattern_10" for the buzzer pattern.

Example

```
<!-- Repeat the sound pattern A three times -->  
<sound pattern="pattern_a" repeat="3" />
```

<command>

Insert the ESC / POS command.

ESC / POS command is specified in hexadecimal encoding.



Refer to the following URL for details of the ESC/POS command.

https://reference.epson-biz.com/modules/ref_escpos/index.php?content_id=2

<layout>

It specifies paper layout. (in ePOS-Device Service Ver.2.2 and later)



Cannot be written in the <page> tag.

Attribute

- type : Specifies the paper type.

Attribute value	Description
"receipt" (default)	Receipt (without black mark)
"receipt_bm"	Receipt (with black mark)
"label"	Die-cut label (without black mark)
"label_bm"	Die-cut label (with black mark)

- width : Specifies paper width (in units of 0.1 mm). Specifies an integer from 290 to 600.
- height : Specifies paper height (in units of 0.1 mm).

Paper Type	Valid value range	Description
Receipt (without black mark)	"0"	Setup not necessary
Receipt (with black mark)	"0" (auto) "284" to "1550" (manual)	Distance from the top of black mark to the top of next black mark
Die-cut label (without black mark)		Distance from the top of label to the top of next label
Die-cut label (with black mark)		Distance from the bottom of black mark to the bottom of next black mark

- margin-top : Specifies top margin (in units of 0.1 mm).

Paper Type	Valid value range	Description
Receipt (without black mark)	"0"	Setup not necessary
Receipt (with black mark)	"-150" to "1500"	Distance from the top of black mark
Die-cut label (without black mark)	"0" to "1500"	Distance from the top of label
Die-cut label (with black mark)	"-15" to "1500"	Distance from the bottom of black mark

- **margin-bottom** : Specifies bottom margin (in units of 0.1 mm).

Paper Type	Valid value range	Description
Receipt (without black mark)	"0"	Setup not necessary
Receipt (with black mark)	"0"	
Die-cut label (without black mark)	"-15" to "0"	Distance from the bottom of label (paper feed direction is a positive number)
Die-cut label (with black mark)	"-15" to "15"	Distance from the top of black mark (paper feed direction is a positive number)

- **offset-cut** : Specifies cut position (in units of 0.1 mm).
In case of die cut label paper, it is a distance from the bottom of label.
When a paper has black mark, it is a distance from the beginning of black mark.

Paper Type	Valid value range	Description
Receipt (without black mark)	"0"	Setup not necessary
Receipt (with black mark)	"-290" to "50"	Distance from the top of black mark to the cutting position
Die-cut label (without black mark)	"0" to "50"	Distance from the bottom of label to the cutting position
Die-cut label (with black mark)	"0" to "50"	Distance from the top of black mark to the cutting position

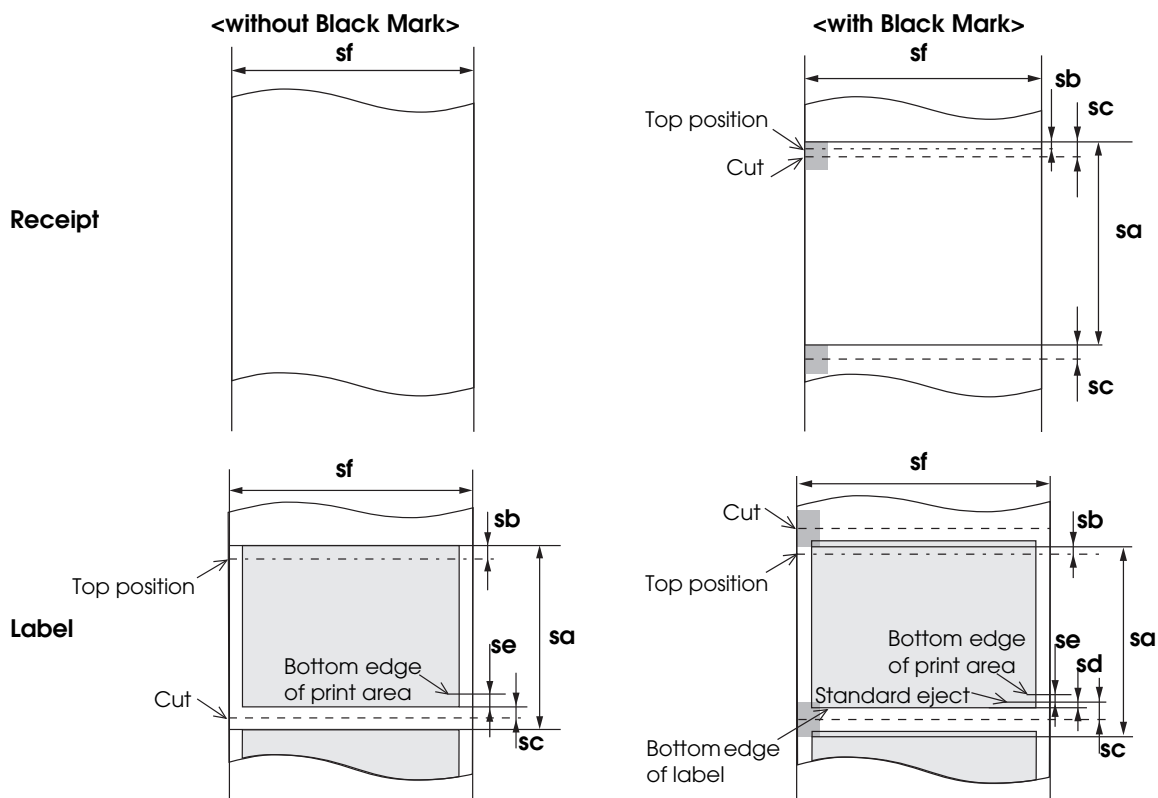
- **offset-label** : Specifies label bottom position (sd) per 0.1 mm unit.

Paper Type	Valid value range	Description
Receipt (without black mark)	"0"	Setup not necessary
Receipt (with black mark)	"0"	
Die-cut label (without black mark)	"0"	
Die-cut label (with black mark)	"0" to "15"	Distance from the top of black mark to the bottom of label

Detailed description

See below for the parameters that can be specified for each type of paper, and the positions for those parameters.

Mark	Parameter
sf	width
sa	height
sb	margin_top
se	margin_bottom
sc	offset_cut
sd	offset_label



Example

```
<!-- To set 58 mm receipt (without black mark) -->
<layout type="receipt" width="580" />
<!-- To set 58 mm receipt (with black mark) -->
<layout type="receipt_bm" width="580" height="0" margin-top="15"
offset-cut="0" />
<!-- To set 58 mm die-cut label (without black mark) -->
<layout type="label" width="580" height="0" margin-top="15"
margin-bottom="-15" offset-cut="25" />
<!-- To set 58 mm die-cut label (with black mark) -->
<layout type="label_bm" width="580" height="0" margin-top="15"
margin-bottom="-15" offset-cut="25" offset-label = "15" />
```

<recover>

Recovers from an error. (in ePOS-Device Service Ver.2.2 and later)

	Cannot be written in the <page> tag.
	Recovers from errors that can be recovered from and clears the buffer. Enable forced transmission mode (p. 85) to use this element.

Example

```
<!-- Recovers from errors that can be recovered from and clears the buffer. -->
<recover />
```

<reset>

Resets the printer. (in ePOS-Device Service Ver.2.2 and later)

	Cannot be written in the <page> tag.
	Other printing commands in the print document are ignored.

Example

```
<!-- Resets the printer -->
<reset />
```

Message Data for the Customer Display



For the display data, refer to [Customer Display Control XML \(p.130\)](#).

<type>display</type>

Request

Sends the display data and setting data to the customer display.

Sub-element	Data type	Description
timeout	int	Sets the time out limit for requests in milliseconds. 1,000 to 60,000 (whole numbers)
displaydata	string	Specifies the customer display control XML. For details, refer to Customer Display Control XML (p.130) .

Example

```
<data>
  <type>display</type>
  <timeout>10000</timeout>
  <displaydata>
    <!-- Customer display control XML -->
    <epos-display xmlns=
      .
    </epos-display>
  </displaydata>
</data>
```

<type>onxmlresult</type>**Response**

Returns the result of sending to the customer display to the application.

Sub-element	Data type	Description
resultdata	string	Receives the response from the data sent to the customer display. For details, refer to <response> (p.131) .

□ Example

```
<data>
  <type>onxmlresult</type>
  <resultdata>
    <response success="true" code="" status="" />
  </resultdata>
</data>
```

Customer Display Control XML

This chapter explains the XML document that controls the customer display.



For customer display message data, refer to Message Data for the [Message Data for the Customer Display \(p.128\)](#).

<epos-display>

This is the XML document sent from the application to the customer display. It requests that the customer display execute the designated function. <epos-display> contains the necessary sub-element for controlling the customer display.

Attribute

❑ xmlns

- We're announcing the epos-display namespace. Find the namespace below.
<http://www.epson-pos.com/schemas/2012/09/epos-display>

Sub-element

Sub-element	Function	Page
<window>	Window settings	90
<text>	Character display	96
<cursor>	Cursor display	98
<blink>	Settings for screen blinking	100
<brightness>	Settings for the customer display brightness	101
<marquee>	Settings for character string marquees	105
<clock>	Time display	110
<clear>	Clears the current window display	111
<reset>	Resets the customer display	112
<command>	Executes the ESC/POS command	113

❑ Example

```
<!-- Example with a text tag added as a sub-element -->
<epos-display xmlns="http://www.epson-pos.com/schemas/2012/09/epos-display">
  <text>Hello world!!</text>
</epos-display>
```

<response>

This is the XML document that returns from the customer display to the application.

Attribute

❑ success

- Acquires the display result.

Attribute value	Description
"true" or "1"	Successfully displayed
"false" or "0"	Failed to display

❑ code

- Acquires the error code.

Attribute value	Description
"EDSP_NOT_FOUND"	Could not find the error code
"EDSP_NOT_OPEN"	Failed to open the device
"EDSP_INVALID_WINDOW"	An unregistered window was designated
"EDSP_SYSTEM_ERROR"	An unexpected error occurred
"EDSP_SCHEMA_ERROR"	The XML structure is invalid
"EX_BADPORT"	An error occurred in internal communication error with the device
"EX_TIMEOUT"	A time-out error occurred in communication with the device
"EX_INVALID_VALUE"	Detected an invalid parameter

❑ status

- "0" is always set.

❑ Example

```
<!-- Successfully performed. -->
<response success="true" code="" status="0"/>
<!-- Failed -->
<response success="false" code="EDSP_NOT_FOUND" status="0"/>
```

<window>

Controls creating and deleting windows and moving the current window.

- ❑ If the attribute number, x, y, width, and height are designated, the designated window is created. When defining the window, define it so that it does not overlap with an already defined window. Specify the size to fit in the customer display size (20 digits x 2 rows).
- ❑ If "true" is designated for attribute number destroy, the designated window is deleted.
- ❑ If an attribute that matches both window creation and deletion is designated, neither process is performed.
- ❑ If an attribute that matches window creation or deletion is not designated, and number is designated, the current window is moved to the designated window.

Attribute

❑ number

- Designates the target window number.

Attribute value	Description
"1"	Window 1
"2"	Window 2
"3"	Window 3
"4"	Window 4

❑ x

- Designates the x coordinate (1 to 20) in a whole number when creating a window.

❑ y

- Designates the y coordinate (1 to 2) in a whole number when creating a window.

❑ width

- Designates the width (1 to 20) in a whole number when creating a window.

❑ height

- Designates the height (1 to 2) in a whole number when creating a window.

□ scrollmode

- Designates the scroll method of the created window when creating a window.

Attribute value	Description
"overwrite" (Default)	When characters are displayed while the display position is at the upper right end, the display position moves to the lower left end. When characters are displayed while the display position is at the lower right end, the position moves to the upper left end.
"v_scroll"	When characters are displayed while the display position is at the upper right end, the display position moves to the lower left end. When characters are displayed while the display position is at the lower right end, the characters already displayed in the lower row are scrolled to the upper row and the lower row is cleared.
"h_scroll"	When characters are displayed while the display position is at the right end, all the characters already displayed in the row where the cursor exists are scrolled one character to the left and a new character is displayed at the right end.

□ destroy

- Designates whether to delete the window designated with number.

Attribute value	Description
"true" or "1"	Executes deletion
"false" or "0" (Default)	Does not execute deletion

Error conditions

The following values are returned in the event of an error.

Error value	Description
"EX_INVALID_VALUE"	When creating a window and the x, y, width, or height overlaps with an already defined window
"EDSP_INVALID_WINDOW"	When moving a window and the designated window does not exist

□ Example

```
<!-- Define window 1 from the top left with width 10, height 2, vertical scroll -->
<window number="1" x="1" y="1" width="10" height="2" scrollmode="v_scroll"/>
<!-- Define window 2 from the center with width 10, height 2, vertical scroll -->
<window number="2" x="11" y="1" width="10" height="2" scrollmode="v_scroll"/>
<!-- Move the current window to window 1 -->
<window number="1"/>
<!-- Delete window 2 -->
<window number="2" destroy="true"/>
```

<text>

Controls display of character strings.

- ❑ If attributes x and y are designated, display is performed from the designated cursor position.
- ❑ If attributes x and y are not designated, display is performed from the current cursor position.

Attribute

❑ x

- Designates the x coordinate for text display (1 to 20) in a whole number

❑ y

- Designates the y coordinate for text display (1 to 2) in a whole number

❑ reverse

- Designates whether to shade displayed characters

Attribute value	Description
"true" or "1"	Performs shading
"false" or "0" (Default)	Does not perform shading

❑ lang

- Designates the language of displayed text. If omitted, the current settings are maintained.

Attribute value	Description
"en" (Default)	English
"ja"	Japanese

❑ Example

```
<!-- Display character strings from top left -->
<text x="1" y="1" lang="ja">Carrots \100</text>
<!-- Display shaded character strings from the current cursor position -->
<text reverse="true">Welcome!!</text>
```

```
<!-- Change the current window to window 2 -->
<window number="2"/>
<text x="1" y="1" lang="ja"> Carrots, $1</text>
```

<cursor>

Controls the cursor location and display settings.

- ❑ Designates the cursor coordinates in the customer display using the x, y designations.
- ❑ Designates the cursor location in the current window using the moveto designation.
- ❑ If x and y and moveto are simultaneously designated, neither is executed.

Attribute

❑ x

- Designates the cursor coordinates in the display using the x, y designations.

❑ y

- Designates the cursor location in the current window using the moveto designation.

❑ moveto

- If x and y and moveto are simultaneously designated, neither is executed.

Attribute value	Description
"top_left"	Moves to the top left
"top_right"	Moves to the top right
"bottom_left"	Moves to the bottom left
"bottom_right"	Moves to the bottom right

❑ type

- Designates the cursor location in the current window.

Attribute value	Description
"none"	Does not display the cursor
"underline"	Displays using an underline format

❑ Example

```
<!-- Move the cursor to the top left of the customer display -->
<cursor x="1" y="1" />
<!-- Move the cursor to the top left of the current window -->
<cursor moveto="top_left" type="underline" />
```

<blink>

Controls screen blinking. Blinking is repeated at the interval designated by interval.



The actual blinking interval is rounded up in 50 ms units.
For example, if interval is 1 to 50, the actual interval is 50 ms, if 51 to 100, then 100 ms.

Attribute

□ interval

- Designates the interval for blinking in a whole number.

Attribute value	Description
"0"	Terminates blinking and sets display to be continuously on
"1" to "12700"	Milliseconds for blinking interval
"12701" to "12750"	Turns off display, maintaining displayed contents

□ Example

```
<!-- Set screen blinking interval to 500 ms -->
<blink interval="500"/>
<!-- Clear screen blinking -->
<blink interval="0"/>
<!-- Turn off screen. Maintain displayed contents -->
<blink interval="12750"/>
<cursor moveto="top_left" type="underline"/>
```

<brightness>

Controls the customer display brightness.

Attribute

□ value

- Designate brightness in percentage.

Attribute value	Description
"20"	Brightness 20%
"40"	Brightness 40%
"60"	Brightness 60%
"100"	Brightness 100%

□ Example

```

<!-- Set brightness to 20% -->
<brightness value="20"/>
<!-- Set brightness to 40% -->
<brightness value="40"/>
<!-- Set brightness to 60% -->
<brightness value="60"/>
<!-- Set brightness to 100% -->
<brightness value="100"/>

```

<marquee>

Controls the marquee for character strings.

Displays the designated character string 1 character at a time at the interval designated in `uwait`. After finishing the complete display, it waits the time designated by `rwait`, and then displays the character string again from the start.

Attribute

☐ format

- Designates the display pattern for the marquee.

Attribute value	Description
"walk" (Default)	Displays the string from the right end of the window.
"place"	Displays the string from the left end of the window.

☐ repeat

- Designates the number of times to repeat the display.

Attribute value	Description
"0" (Default)	Repeats indefinitely
"1" to "127" (whole number)	Repeats the designated number of times

☐ uwait

- Designates the wait interval for each character display in milliseconds.

Attribute value	Description
"0" to "2000" (whole number)	Wait interval for each character display

☐ rwait

- Designates the wait interval after displaying the end of the character string in milliseconds.

Attribute value	Description
"100" to "2000" (whole number)	Wait period for after displaying the end of the character string

☐ lang

- Sets the display language. If omitted, the current settings are maintained.

Attribute value	Description
"en" (Default)	English
"ja"	Japanese

☐ Example

```
<!-- Display marquee from right of the screen at 200 ms/character, repeat at 1000 ms intervals -->
<marquee uwait="200" rwait="1000">Welcome</marquee>
<!-- Display marquee from left of the screen at 200 ms/character, repeat at 1000 ms intervals -->
<marquee uwait="200" rwait="1000" format="place">Welcome!!</marquee>
```

<clock>

Displays the time in the bottom right corner of the customer display screen.
The local time managed by the TM-Intelligent's OS is displayed.

❑ Example

```
<clock/>
```

<clear>

Clears the display of the current window.
If a window does not exist, the whole display is cleared.

❑ Example

```
<clear/>
```

<reset>

Resets the display. Resetting the customer display sets it to the following conditions.

- ❑ All character strings are cleared, and all registered windows are discarded.
- ❑ Returns the cursor to the default conditions and moves it to the customer display origin point.
- ❑ Returns the display blinking and display brightness to default conditions.

❑ Example

```
<reset/>
```

<command>

Executes any ESC/POS command.
Designate a command with a hexadecimal character string.



For details on ESC/POS, see the specifications for this product. You need a contract separately to acquire the specifications. For details, contact the store of your purchase.

❑ Example

```
<command>01025AEF3B405C</command>
```

Message Data for the POS Keyboard

<type>onkeypress</type>

Response

Returns the input data from the POS keyboard to the application.

Sub-element	Data type	Description
keycode	int	Receives the key code decimals input with the POS keyboard. For key code details, refer to the List of KeyCode (p.189)
ascii	string	Receives the characters created by key codes. Characters converted into symbols and capital letters from holding down the shift key are set.

Character strings in received data

Escape processing is performed for the following character strings.

Character string before replacement	After replacement
& (Ampersand)	&
> (Greater-than symbol)	>
< (Lesser-than symbol)	<
"(Double quotation mark)	"
' (Single quotation mark)	'

❑ Example

```
<data>
<type>onkeypress</type>
<keycode>49</keycode>
<ascii>a</ascii>
</data>
```

<type>setprefix</type>

Designates the key code to determine the start of character strings to detect input from the POS keyboard. If a key code designated by this message is input, the character string from then until the Enter key is pressed is notified as the onstring message. This is used, for example, when using the POS keyboard for bar-code alternate input. To stop the onstring message, send a setprefix message without a keycodes.

Request

Designates the key code to determine the start of character strings to detect input from the POS keyboard.

Sub-element	Data type	Description
keycodes	int	Designates the key code decimals to determine the start of the character string. For key code details, refer to the List of KeyCode (p.189) .

□ Example

```
<data>
  <type>setprefix</type>
  <keycodes array="true">49</keycodes>
  <keycodes>50</keycodes>
  <keycodes>51</keycodes>
  <keycodes>52</keycodes>
</data>
```

<type>onstring</type>

Response

Returns the input data from the POS keyboard to the application.

Notifies you that input from one of the key codes designated with setprefix to Enter is detected.

The detected key code information, as well as the sequence of input character strings, is acquired in the argument.

Sub-element	Data type	Description
input	string	Receives the input character strings. Does not include the key to determine the start of the character string or the Enter key.
prefix	int	Receives the key code that determined the start of the character string. For key code details, refer to the List of KeyCode (p.189) .

Character strings in received data

Escape processing is performed for the following character strings.

Character string before replacement	After replacement
& (Ampersand)	&
> (Greater-than symbol)	>
< (Lesser-than symbol)	<
"(Double quotation mark)	"
' (Single quotation mark)	'

❑ Example

```
<data>
<type>onstring</type>
<input>&amp;2398749238429</input>
<prefix>49</prefix>
</data>
```

<type>setMSRPrefix</type>

Specifies the key code to determine the card information reception with a keyboard with an MSR.

If the key code specified by this message was input, the character strings of the card information received from there will be notified as an ondata message.

Request

Specifies the key code to determine the card information reception with a keyboard with an MSR.

Sub-element	Data type	Description
keycode	int	Sets the keycode for determining the beginning when card information is received. For key code details, refer to the List of KeyCode (p.189) .

❏ Example

```
<data>
  <type>setMSRPrefix</type>
  <keycode>52</keycode>
</data>
```

<type>ondata</type>

Response

Returns the received data of the card information from the keyboard with an MSR to the application.

Sub-element	Data type	Description
track1	string	Entire data of JIS1 track1
track2	string	Entire data of JIS1 track2
track4	string	Entire data of JIS2 track
account_number	string	PAN of JIS1 track2
expiration_date	string	Expiration date of JIS1 track2
surname	string	Surname
first_name	string	First name
middle_initial	string	Middle name/initial
title	string	Title
service_code	string	Service code
track1_dd	string	Desired data of JIS1 track1
track2_dd	string	Desired data of JIS1 track2

Character strings in received data

Escape processing is performed for the following character strings.

Character string before replacement	After replacement
& (Ampersand)	&
> (Greater-than symbol)	>
< (Lesser-than symbol)	<
"(Double quotation mark)	"
' (Single quotation mark)	'

□ Example

```
<data>
<track1>B4205296141577006^EPSON/TARO          ^160410100000      00573000000</track1>
<track2>9999296999977006=16041010000057300000</track2>
<track4>$600003046999929699997700680016044500001604000000000000000000005730</track4>
<account_number>9999296999977006</account_number>
<expiration_date>1604</expiration_date>
<surname>EPSON</surname>
<first_name>TARO</first_name>
<middle_initial>          </middle_initial>
<title></title>
<service_code>101</service_code>
<track1_dd>00000      00573000000</track1_dd>
<track2_dd>0000057300000</track2_dd>
</data>
```

Message Data for the Serial Device



This message data cannot be used with a TM-i Series printer with the specification of "without a Serial port".

<type>sendcommand</type>

Request

Sends commands to the serial device.

Sub-element	Data type	Description
command	string	Hexadecimal expressed character string for the command string

Example

```
<data>
  <type>sendcommand</type>
  <command>1B123344FF1F5D3C</command>
</data>
```

<type>oncommandreply</type>



Response data from the device may be sent split up. If so, oncommandreply events will occur multiple times. Use the oncommandreply events to check that the whole of the response data has been received. If there is data still to be received, wait for it to arrive before proceeding with the processing.

Response

Returns the data from the serial device to the application.

Sub-element	Data type	Description
status	string	Data reception status For details, refer to the status below.
data	string	Hexadecimal expressed character string for the data string

❑ status

Element value	Description
"SUCCESS"	Successfully received data

❑ Example

```
<data>
<type>oncommandreply</type>
  <status>SUCCESS</status>
  <data>1B123344FF1F5D3C</data>
</data>
```

Device Control Script

This section describes the information that the user will need in order to develop device control script.

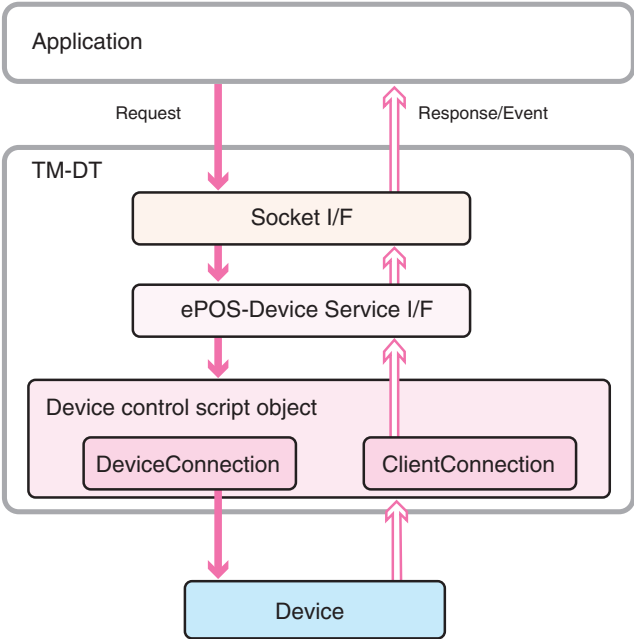


Cannot be used with TM-i Series.

Programming

Using Device Control Script

Using API for the device control script provided by ePOS-Device Service enables customization of device data processing and use of a new device from applications.



When "open_device" message (p.60) is sent, an object is created so that the ePOS- Device Service I/F on TM-DT can use the device control script supported by the requested device. The device can be controlled with the created object.

Device Control Script Object

The ePOS-Device Service I/F passes the objects shown below to the device control script. Using these objects enables the device control script to communicate the application and device.

Object	Description
ClientConnection	Object used to send data to a device object on the application.
DeviceConnection	Object used to send/receive data to/from the device.

Functions Using Device Control Script Object

Using API for the device control script enables the following functions:

- ❑ Calling a desired event of device object on the browser
- ❑ Sending data to the device
- ❑ Receiving data generated in the device

Configuration of Device Control Script

Coding of the device control script must satisfy the following conditions:

- ❑ Describe the codes required for device control collectively in a file.
(Only one file can be registered when using WebConfig for device setting.)
- ❑ The part before the first dot "." of the file name must be identical to the constructor name.
(Example) File name : Keyboard_Generic.ver1.0.js -> Constructor name : Keyboard_Generic
- ❑ For external reference of constructor, declare exports.
Ex) `exports.Keyboard_Generic = Keyboard_Generic;`
- ❑ The constructor must have two arguments.
- ❑ The device control script must have the properties as shown below. Set an appropriate name with the constructor.
 - DEVICE_TYPE property (Object type :String)

Set value	Description
type_keyboard	Specify this to use a keyboard device.
type_scanner	Specify this to use a barcode scanner.
type_simple_serial	Specify this to perform simple serial communication.

- DEVICE_GROUP property (Object type :String)

Set value	Description
group_hid	Specify this to use an input device that can run with an HID driver.
group_serial	Specify this to use a serial communication device.
group_other	Specify this to use other device.

- ❑ The "onDeviceData" method is provided to receive data generated in the device.
For details, refer to [Device Control Script Name object \(p.150\)](#).
- ❑ The method corresponding to the method of a device object that can run on the application is provided. For details, refer to [Any event \(p.154\)](#).

Configuration Example of Device Control Script

```
exports.Keyboard_Generic = Keyboard_Generic;           // Exports declaration

function Keyboard_Generic(clientConn, deviceConn){      // Having two arguments
                                                         // and the same name as the
                                                         // file name
    this.DEVICE_TYPE = 'type_keyboard';                // Having the
                                                         // "DEVICE_TYPE" property
    this.DEVICE_GROUP = 'group_hid';                   // Having the
                                                         // "DEVICE_GROUP" property

    this.clientConn = clientConn;
    this.deviceConn = deviceConn;
    .....
    .....
}

Keyboard_Generic.prototype = {
    onDeviceData : function(event, keycode, ascii){...}, // Having the
                                                         // "onDeviceData" method
    setprefix : function(data){...}                     // Having the method
                                                         // corresponding to the
                                                         // device object
}
```

List of Device Control Script API

The device control script API provides the following objects:

- ❑ ClientConnection object ([p.150](#))
- ❑ DeviceConnection object ([p.150](#))
- ❑ Device control script name object ([p.150](#))

ClientConnection object

Element	API	Description	Page
Method			
	send	Sends data to a device object that can run with a browser.	151

DeviceConnection object

Element	API	Description	Page
Method			
	send	Sends data to a serial communication device.	152

Device Control Script Name object

Element	API	Description	Page
Event			
	onDeviceData	Key input receipt event	153
	onDeviceData	Data receipt event from a serial communication device	153
	Any event	API execution result receipt event of a device object that runs with a browser	154

ClientConnection Object

The object passed to the first parameter of the constructor of the device control script.

send

Sends data to a device object that can run with a browser.

Syntax

```
send(event, data);
```

Parameter

- event :Object type : (String)
Specifies the event name of the device object.
- data :Object type : (Object)
Specifies the data to be passed to the event of the device object.

Example

In the example below, the "onkeypress" event of the device object is called and 49 is obtained from data.keycode and 'l' from data.ascii using the "data" parameter of the "onkeypress" event.

```
data = {'keycode' : 49, 'ascii' : 'l'};  
clientConn.send('onkeypress', data)
```

DeviceConnection Object

The object passed to the second parameter of the constructor of the device control script.

send

Sends data to a serial communication device.



No data is sent when an input device that can run with an HID driver is used.

Syntax

```
send(data) ;
```

Parameter

- data :Object type : (Buffer)
Specifies the data to be sent to the device.

Device Control Script Name Object

onDeviceData event (Input Device That Can Run with a HID Driver)

The event used to receive the data detected from a key input device.

For the device control script for a key input device, describe an event in this format.

Syntax

```
onDeviceData(event, keycode, ascii);
```

Parameter

- event** :Object type : (Number)
 Receives a direction of key operation.

Value	Description
1	Key down
2	Key up
- keycode** :Object type :(Number)
 Receives a keycode. For details, refer to [List of KeyCode \(p.189\)](#).
- ascii** :Object type : (String)
 Receives a character corresponding to the operated key.
 "undefined" is applied when no characters corresponding to the keycode (F1 key, for example) exist.

onDeviceData event (Serial Communication Device)

The event used to receive the data detected from a serial communication device.

For the device control script for a serial communication device, describe an event in this format.

Syntax

```
onDeviceData(data) ;
```

Parameter

- data** :Object type : (Buffer)
 Receives the data received from a serial communication device.

Any event

The event used to receive the API execution results of the device object that can run with a browser.

Syntax

Name specified with callEvent(data) ;

Parameter

- data :Object type : (Object)
Receives the object specified with callEvent of the device object.

Sample Program

This chapter describes the sample program.



A sample program for an Android/ iOS environment is provided in the ePOS-Device XML.

Outline

The sample program provides the function to scan data with the barcode scanner and print this data.

<Android>

ePOS-DeviceXML Sample

IP Address 192.168.192.168

DeviceID Printer local_printer

DeviceID Scanner local_scanner

disconnect

OnData

1000000001044

1000000001037

1000000001020

1000000001013

1000000001051

print

Print success.

local_scanner : OK

local_printer : OK

Connect to server success.

<iOS>

ePOS-DeviceXML Sample

IPAddress 192.168.192.168

DeviceID Printer local_printer

DeviceID Scanner local_scanner

disconnect

OnData:

1000000001105

1000000001082

1000000001020

1000000001051

1000000001099

print

Print succeeded.

local_scanner : OK

local_printer : OK

Connect to server success.

Building Environment for Android

Environment

Device

- ☐ TM-Intelligent
 - TM-DT Series
 - TM-i Series (TM-i firmware Ver.4.0 or later)
- ☐ Barcode Scanner

Execution environment

This section explains the following environments. Information such as the webpage URLs and versions of files to download is current as of February, 2013. Read the information you need depending on your environment.

- ☐ Java SE Development Kit 6 Update 37
- ☐ Eclipse Classic 4.2.1
- ☐ Android SDK Tools
- ☐ ADT Plugin 20.0.3
- ☐ Sample Program(ePOS-Device_Sample_XML_Vx.x.x.zip)

Flow of environment construction for TM-DT Series

The following illustrates the flow of constructing the environment for the sample program.



If the Windows Security screen appears while using EPSON TMNet WebConfig, enter your user name and password. The default values are as follows:

User name : epson

Password : epson

1. Installing Java SE Development Kit (p.161)

Download and install Java SE Development Kit on the computer for setting.



2. Setting Up Eclipse (p.163)

Construct Eclipse on the computer for setting.



3. Setting Up Android SDK Manager (p.165)

Construct Android SDK Manager on the computer for setting.



4. Installing ADT Plugin (p.170)

Install the ADT Plugin on Eclipse.



5. Importing the Sample Program (p.174)

Import the sample program to Eclipse.



6. Network Settings

Configure the system and TM-DT network settings. For the TM-DT network settings, use either of the following procedures:

- Configuring the Windows network settings with TM-DT
- Installing TMNet WinConfig in an external device (Windows computer) to be used for configuration





7. Connection of barcode scanner to TM-DT

Connect a barcode scanner to TM-DT.



8. [Setting the Barcode Scanner \(p.176\)](#)

Register the barcode scanner to be connected into TM-DT. Make registration using an EPSON TMNet WebConfig.

Flow of environment construction for TM-i Series

The following illustrates the flow of constructing the environment for the sample program.



If the Windows Security screen appears while using EPSON TMNet WebConfig, enter your user name and password. The default values are as follows:

User name : epson

Password : epson

1. Installing Java SE Development Kit (p.161)

Download and install Java SE Development Kit on the computer for setting.

2. Setting Up Eclipse (p.163)

Construct Eclipse on the computer for setting.

3. Setting Up Android SDK Manager (p.165)

Construct Android SDK Manager on the computer for setting.

4. Installing ADT Plugin (p.170)

Install the ADT Plugin on Eclipse.

5. Importing the Sample Program (p.174)

Import the sample program to Eclipse.

6. Connection of barcode scanner to TM-i

Connect a barcode scanner to TM-i.



7. Network Settings

Make the network settings for the TM-i, using EPSON TMNet WebConfig.



8. [Setting the Barcode Scanner \(p.176\)](#)

Register the barcode scanner to be connected into TM-i. Make registration using an EPSON TMNet WebConfig.

Installing Java SE Development Kit

Download and install JDK (Java Development Kit).

Downloading JDK

- 1 Access the following URL and download Java SE 6 JDK.
<http://www.oracle.com/technetwork/java/javase/downloads/index.html>

Java SE 6 Update 37
This release addresses security concerns. Oracle strongly recommends that all Java SE 6 users upgrade to this release. [Learn more](#) ▶

JDK
DOWNLOAD ▶

JDK 6 Docs

- Installation Instructions
- ReadMe
- ReleaseNotes
- Oracle License
- Java SE Products
- Third Party Licenses
- Certified System Configurations

JRE
DOWNLOAD ▶

JRE 6 Docs

- Installation Instructions
- ReadMe
- ReleaseNotes
- Oracle License
- Java SE Products
- Third Party Licenses
- Certified System Configurations

- 2 The Java SE Downloads window appears. Select (Accept License Agreement) and select the file to be installed. Save the file to a desired location.
(In this manual, download Windows x86 to construct the environment.)

Overview Downloads Documentation Community Technologies Training

Java SE Downloads

Thank you for downloading this release of the Java™ Platform, Standard Edition Development Kit (JDK™). The JDK is a development environment for building applications, applets, and components using the Java programming language.

The JDK includes tools useful for developing and testing programs written in the Java programming language and running on the Java™ platform.

Looking for the JavaFX SDK?
The JavaFX SDK is available [here](#)

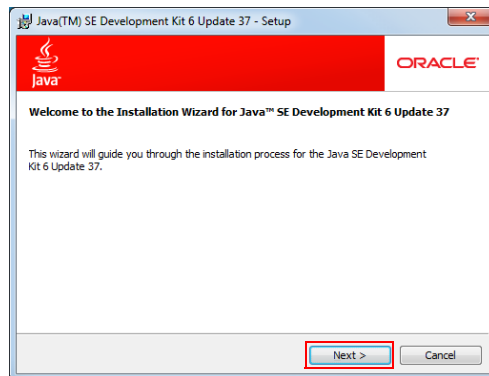
Java SE Development Kit 6 Update 37
You must accept the Oracle Binary Code License Agreement for Java SE to download this software.

☒ Accept License Agreement ☐ Decline License Agreement

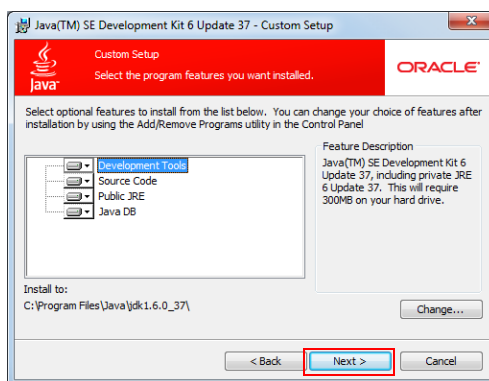
Product / File Description	File Size	Download
Linux x86	65.43 MB	jdk-6u37-linux-i586-rpm.bin
Linux x86	68.44 MB	jdk-6u37-linux-i586.bin
Linux x64	65.65 MB	jdk-6u37-linux-x64-rpm.bin
Linux x64	68.71 MB	jdk-6u37-linux-x64.bin
Linux Intel Itanium	53.95 MB	jdk-6u37-linux-ia64-rpm.bin
Linux Intel Itanium	60.67 MB	jdk-6u37-linux-ia64.bin
Solaris x86	68.35 MB	jdk-6u37-solaris-i586.sh
Solaris x86	119.94 MB	jdk-6u37-solaris-i586.tar.Z
Solaris SPARC	73.36 MB	jdk-6u37-solaris-sparc.sh
Solaris SPARC	124.71 MB	jdk-6u37-solaris-sparc.tar.Z
Solaris SPARC 64-bit	12.13 MB	jdk-6u37-solaris-sparcv9.sh
Solaris SPARC 64-bit	15.42 MB	jdk-6u37-solaris-sparcv9.tar.Z
Solaris x64	8.45 MB	jdk-6u37-solaris-x64.sh
Solaris x64	12.18 MB	jdk-6u37-solaris-x64.tar.Z
Windows x86	69.72 MB	jdk-6u37-windows-i586.exe
Windows x64	59.73 MB	jdk-6u37-windows-x64.exe

Installing JDK

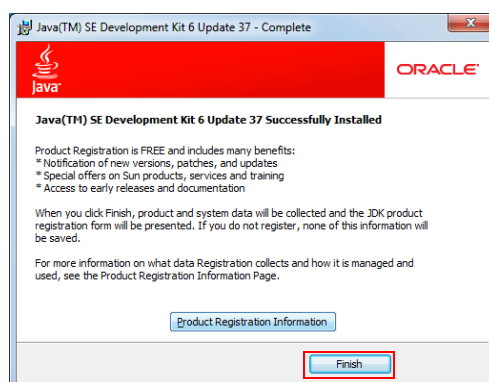
- 1 Start the downloaded file. The following window appears. Click the (Next) button.



- 2 Select the function to install and click the (Next) button. To change the installation destination, click the (Change) button and change the destination. (In this manual, the default destination is used.)



- 3 Installation starts.
- 4 When installation completes, the following window appears. Click the (Finish) button.



Setting Up Eclipse

Download, install and set up Eclipse.

Downloading Eclipse

Access the following URL and download Eclipse. Save the file to a desired location.

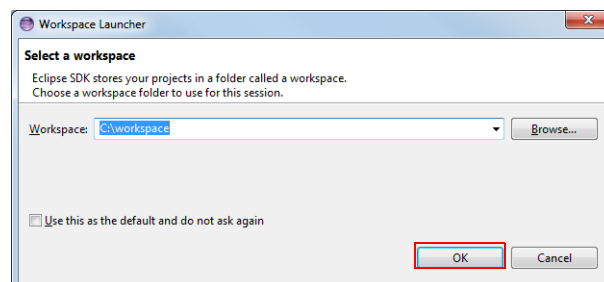
(In this manual, download and set up Eclipse Classic)

<http://www.eclipse.org/downloads/>



Installing Eclipse and Configuring Initial Settings

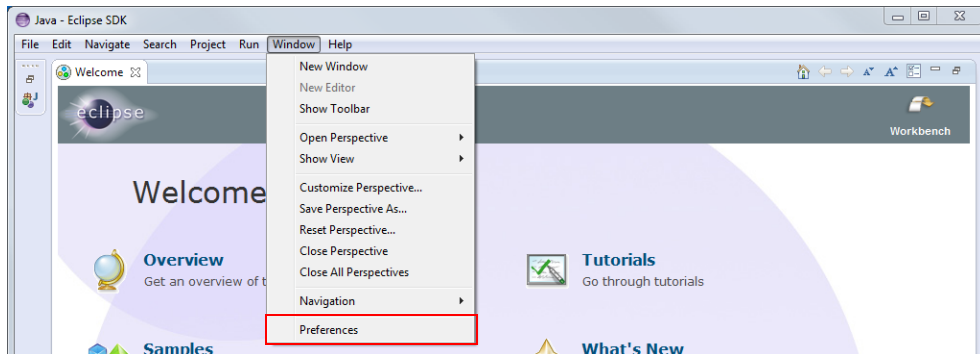
- 1 Extract the downloaded file to a desired location.
(In this manual, extract the file under "C:\".)
- 2 Double-click "eclipse.exe" to start Eclipse.
- 3 Set up the work folder during Eclipse startup process. Click the (OK) button.
(In this manual, the default settings are used.)



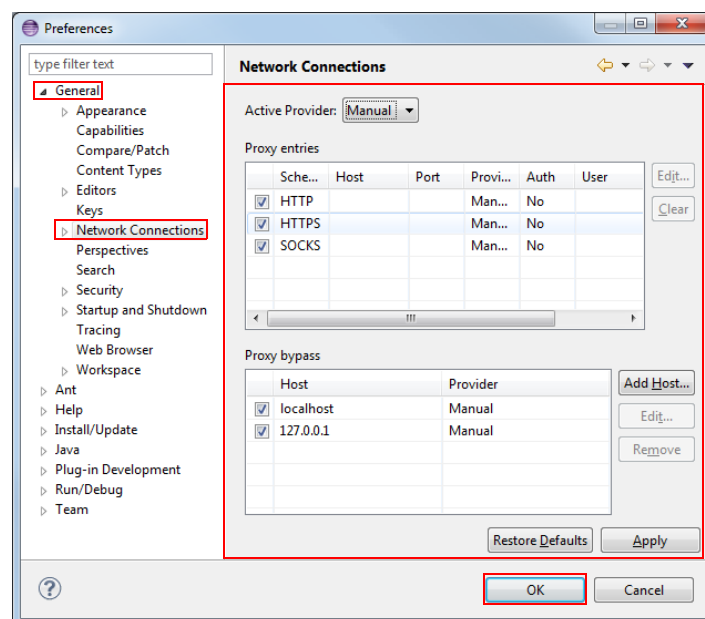
Setting Proxy for Eclipse

Use this operation when proxy setting is required for Internet connection. If not required, this operation is unnecessary.

- 1 Select (Window) - (Preferences).



- 2 Select (General)-(Network Connections) from the left column. Set the proxy and click the (OK) button.



Setting Up Android SDK Manager

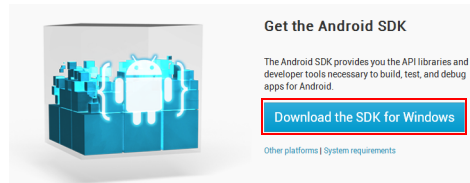
Download and install Android SDK Manager. Create Android Virtual Device.

Downloading Android SDK Manager

Access the following URL and download Android SDK Manager.

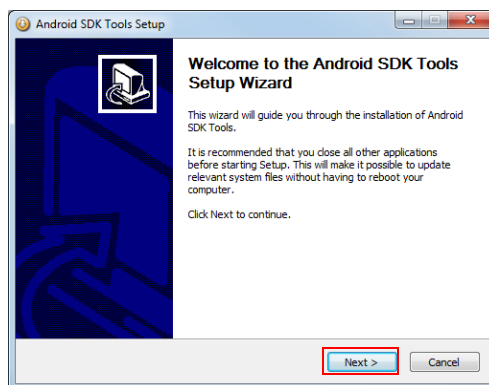
(In this manual, download the installer for Windows.)

<http://developer.android.com/sdk/index.html>



Installing Android SDK Manager

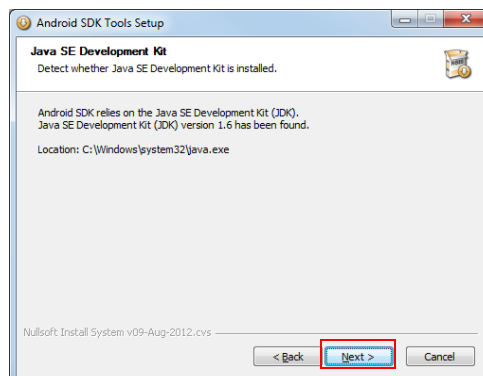
- 1 Start the downloaded file. The following window appears. Click the (Next) button.



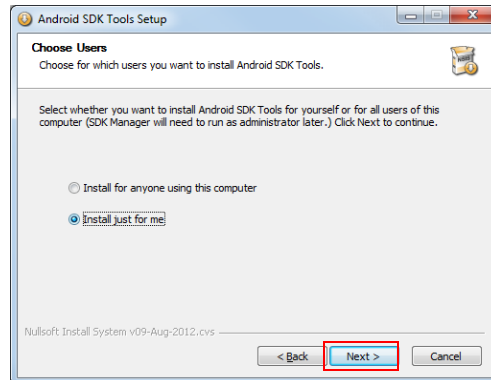
- 2 Click the (Next) button.



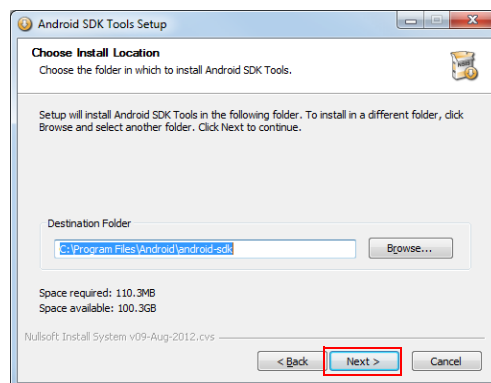
If JDK is not installed, it is notified at this point.



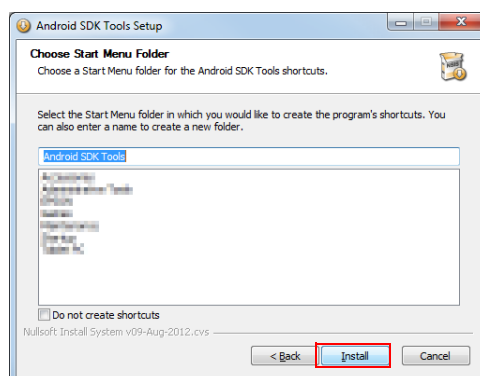
- 3** Specify the user destination. Click the (Next) button.
(In this manual, the default settings are used.)



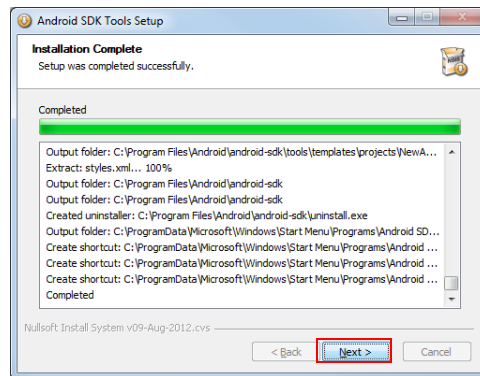
- 4** Specify the Install location destination. Click the (Next) button.
(In this manual, the default settings are used.)



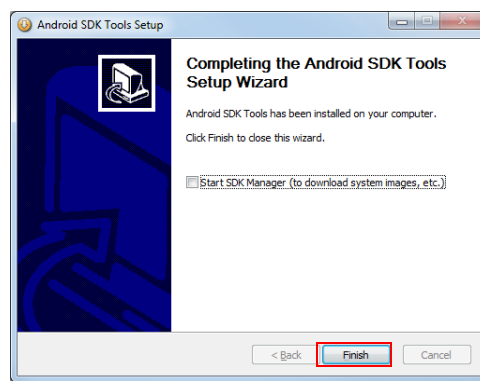
- 5** Specify the name to be registered to Start Menu. Click the (Install) button.



- 6 When installation completes, click the (Next) button.



- 7 Clear the (Start SDK Manager) checkbox. Click the (Finish) button.



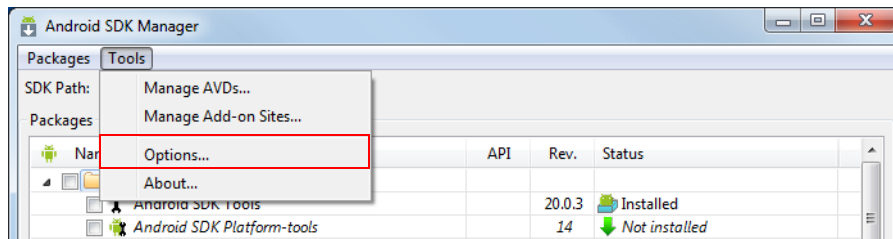
Setting Android SDK Manager

- 1 Start Android SDK Manager.
(Start)-(All Programs)-(Android SDK Tools)-(SDK Manager)

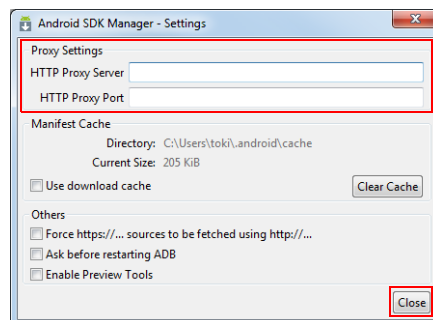


On Windows Vista or later, it is required to run Android SDK Manager as the administrator. To run as the administrator, right-click Android SDK Manager and select [Run as administrator]. If Android SDK Manager is not run as the administrator, package download will fail.

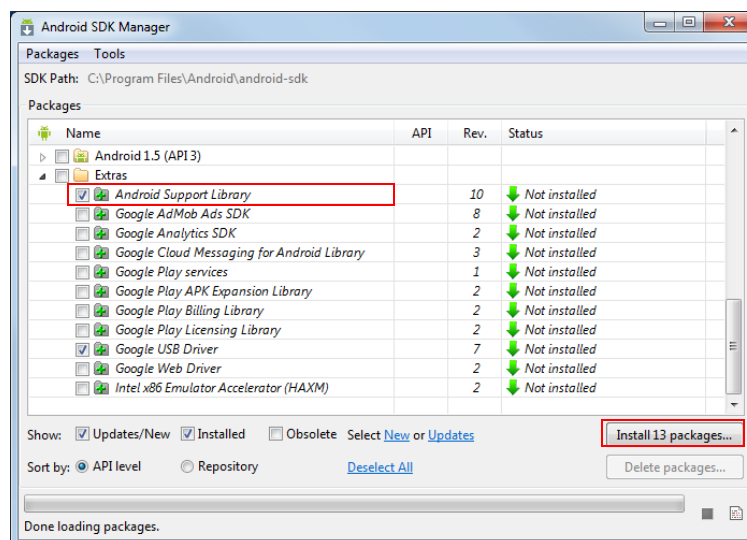
- 2 When the proxy setting is required, select (Tools)-(Options).



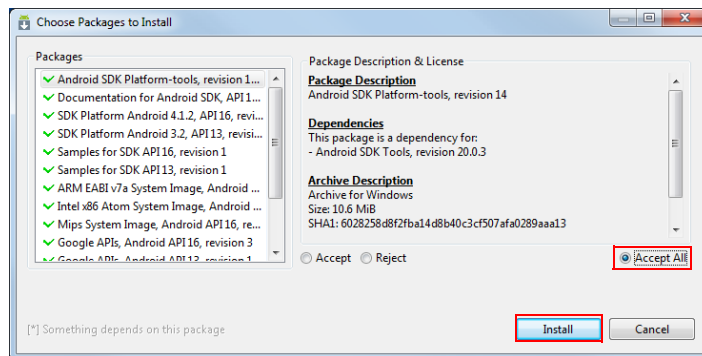
- 3 The proxy setting window appears. Set the proxy and click the (Close) button.



- 4 Select the version of the device to develop and (Android Support Library) inside (Extras), then click (Install XX packages...).



- 5 Select (Accept All) and click the (Install) button. Installation of Virtual Device starts.



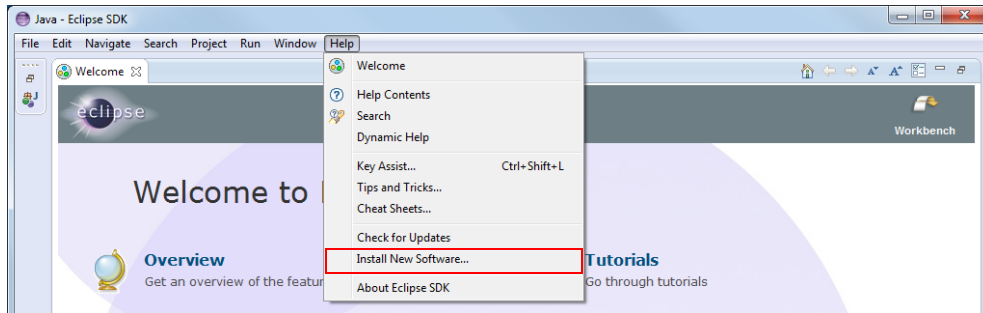
Installing ADT Plugin

install ADT (Android Development Tools) Plugin.

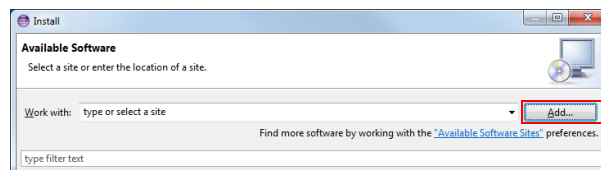


To execute this operation, the computer must be connected to the Internet. If the proxy setting is required for Internet connection, refer to [Setting Proxy for Eclipse \(p.164\)](#).

- 1 Start Eclipse. Select (Help)-(Install New Software).

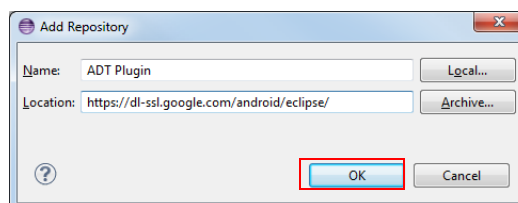


- 2 Click the (Add) button.

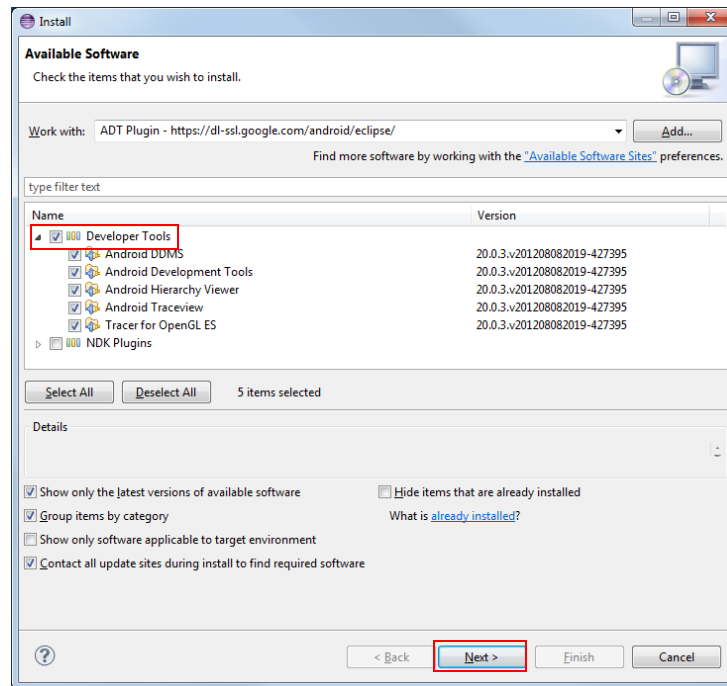


- 3 Enter (Name) and (Location), and click the (OK) button.

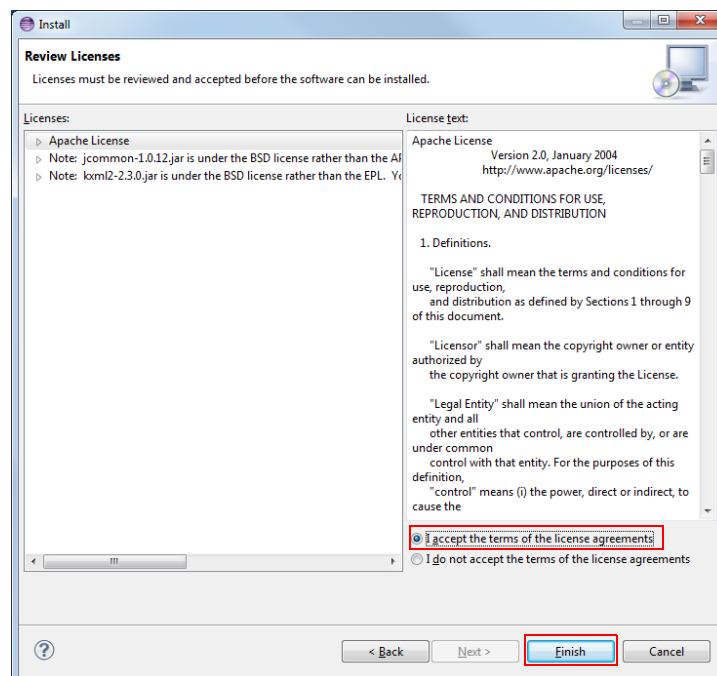
Item	Description
Name	ADT Plugin
Location	https://dl-ssl.google.com/android/eclipse/



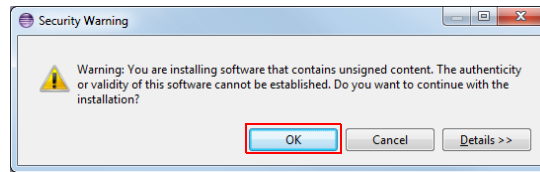
- 4 Select the (Developer Tools) checkbox and click the (Next) button.



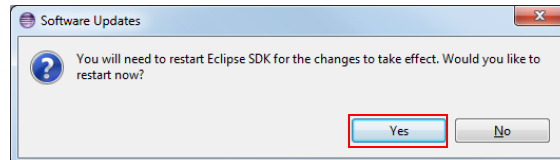
- 5 Confirm the plugin to be installed, and click the (Next) button.
- 6 After confirming the terms of the license agreement, select the (I accept...) checkbox and click the (Finish) button.



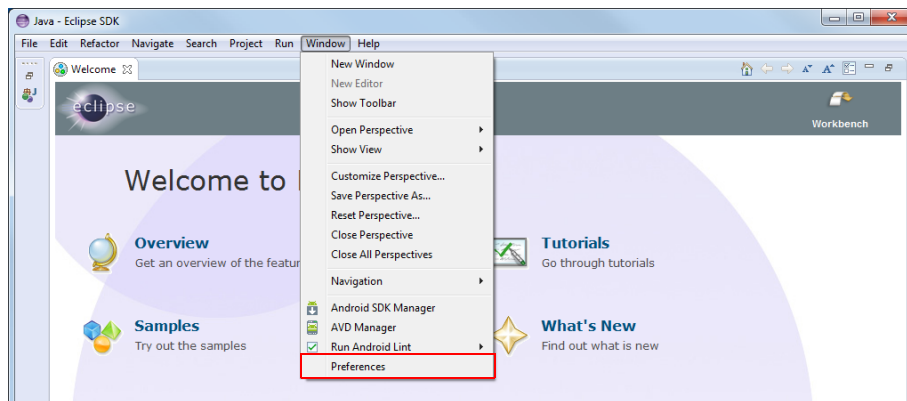
- 7** Installation starts. If the following window appears, click the (OK) button.



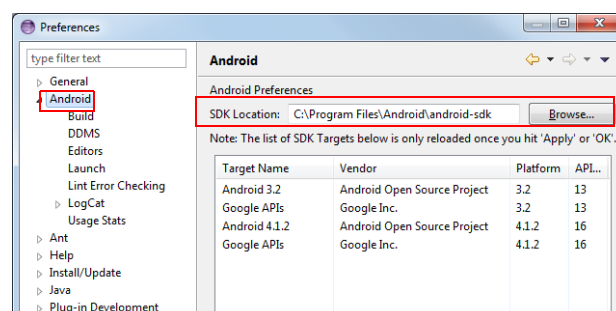
- 8** Click the (Yes) button to restart Eclipse.



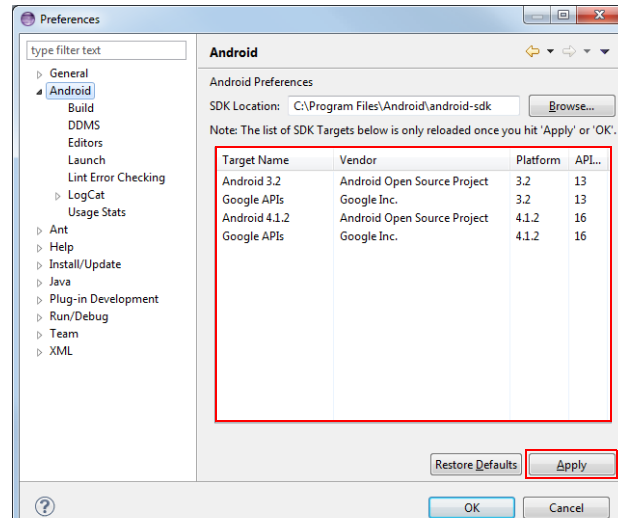
- 9** Select (Window) - (Preferences).



- 10** Select (Android) from the left column. Click the (Browse...) button to display the Android folder where SDK Manager is installed.
(In this manual, display "C:\Program Files\Android\android-sdk".)



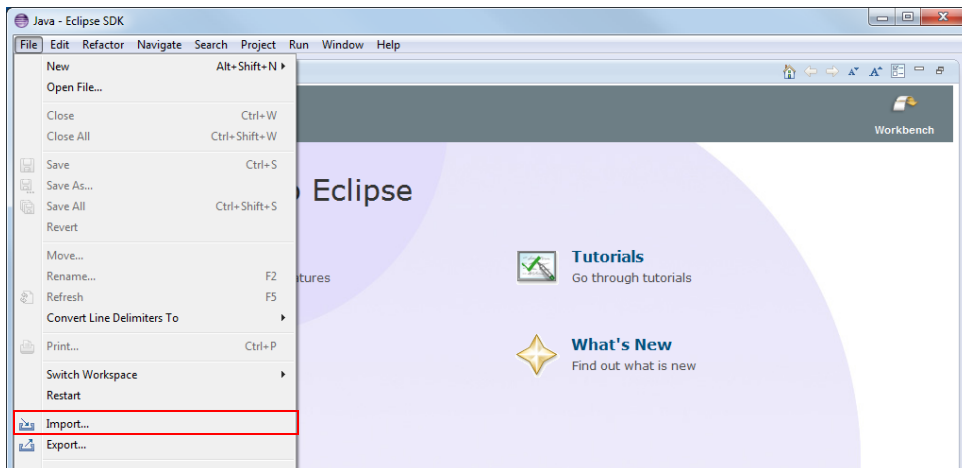
11 Click the (Apply) button. Confirm that the installed package is displayed.



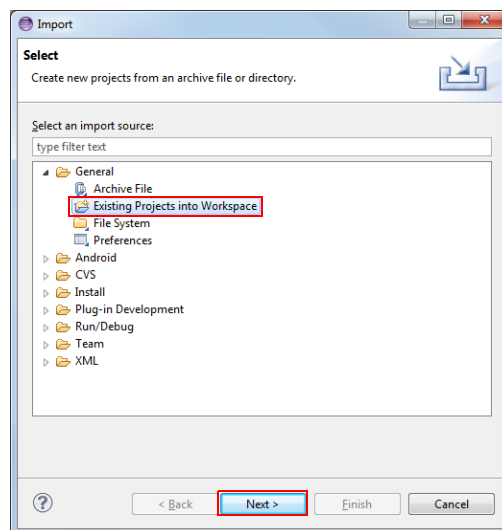
Importing the Sample Program

Import the sample program to Eclipse.

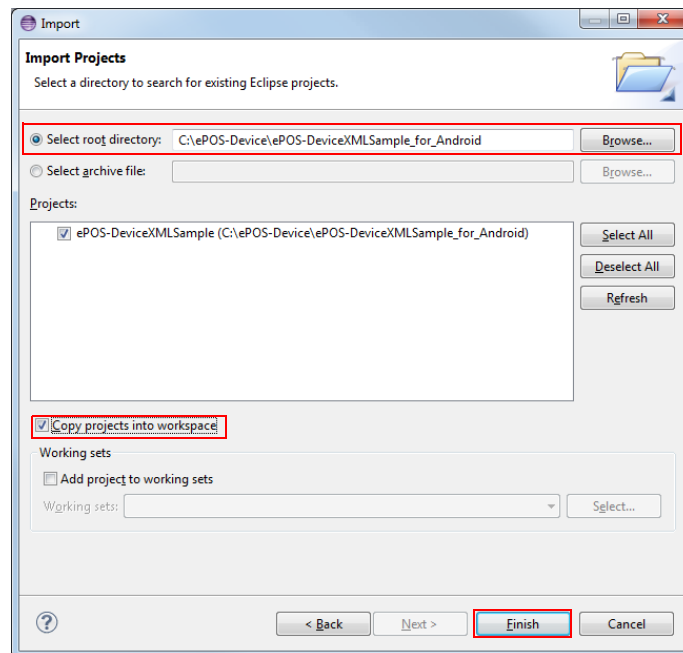
- 1 Extract ePOS-Device_Sample_XML_Vx.x.x.zip, and save it to any folder.
- 2 Start Eclipse. Select (File)-(Import).



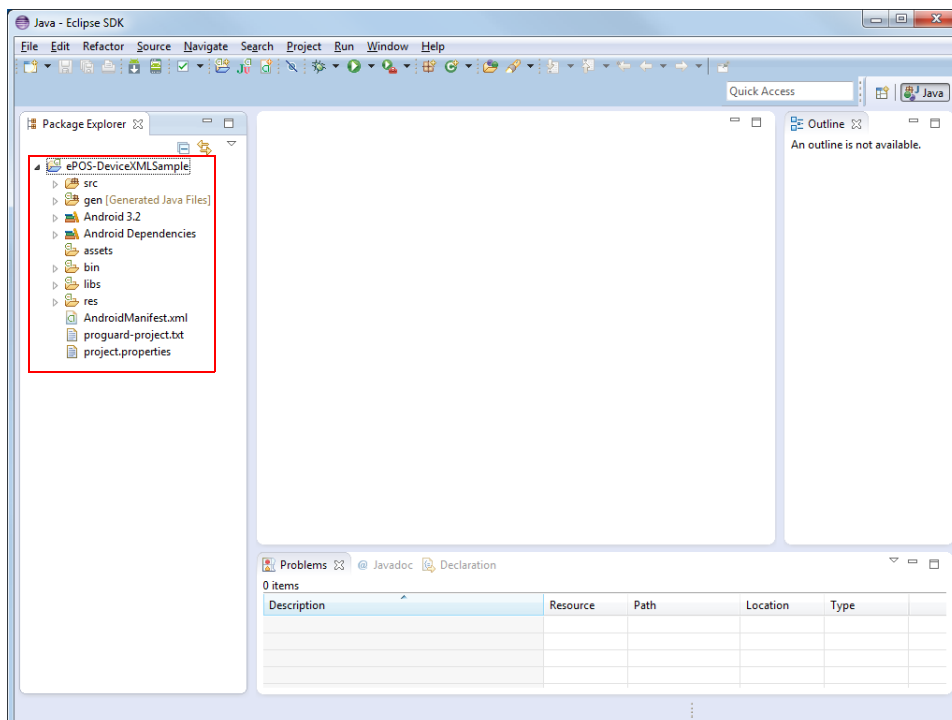
- 3 Select (General)-(Existing Projects into Workspace). Click the (Next) button.



- 4 Click the (Browse...) button to display the saved sample program. Select the (Copy projects into workspace) checkbox and click the (Finish) button.



- 5 Confirm that the application has been properly imported.



Setting the Barcode Scanner

Register the barcode scanner to be controlled by ePOS-Device XML into TM-Intelligent. Configure the settings using EPSON TMNet WebConfig.

EPSON TMNet WebConfig can be launched from your web browser.



For details on settings used, refer to the Technical Reference Guide for each TM-Intelligent.

Set the following.

Setting	Setting Value
Device ID	local_scanner
Device name	Select the relevant device from the list.
Control Script	Scanner_Generic.js

Building Environment for iOS

Environment

Device

- ☐ TM-Intelligent
 - TM-DT Series
 - TM-i Series (TM-i firmware Ver.4.0 or later)
- ☐ Barcode Scanner

Execution environment

This section explains the following environments. Information such as the webpage URLs and versions of files to download is current as of February, 2013. Read the information you need depending on your environment.

- ☐ Mac OS X 10.7.5
- ☐ Apple Xcode 4.5.2
- ☐ iOS SDK 6.0
- ☐ Sample Program(ePOS-Device_Sample_XML_Vx.x.x.zip)

Flow of environment construction for TM-DT Series

The following illustrates the flow of constructing the environment for the sample program.



If the Windows Security screen appears while using EPSON TMNet WebConfig, enter your user name and password. The default values are as follows:

User name : epson

Password : epson

1. Network Settings

Configure the system and TM-DT network settings. For the TM-DT network settings, use either of the following procedures:

- Configuring the Windows network settings with TM-DT
- Installing TMNet WinConfig in an external device (Windows computer) to be used for configuration



2. Connection of barcode scanner to TM-DT

Connect a barcode scanner to TM-DT.



3. [Setting the Barcode Scanner \(p.180\)](#)

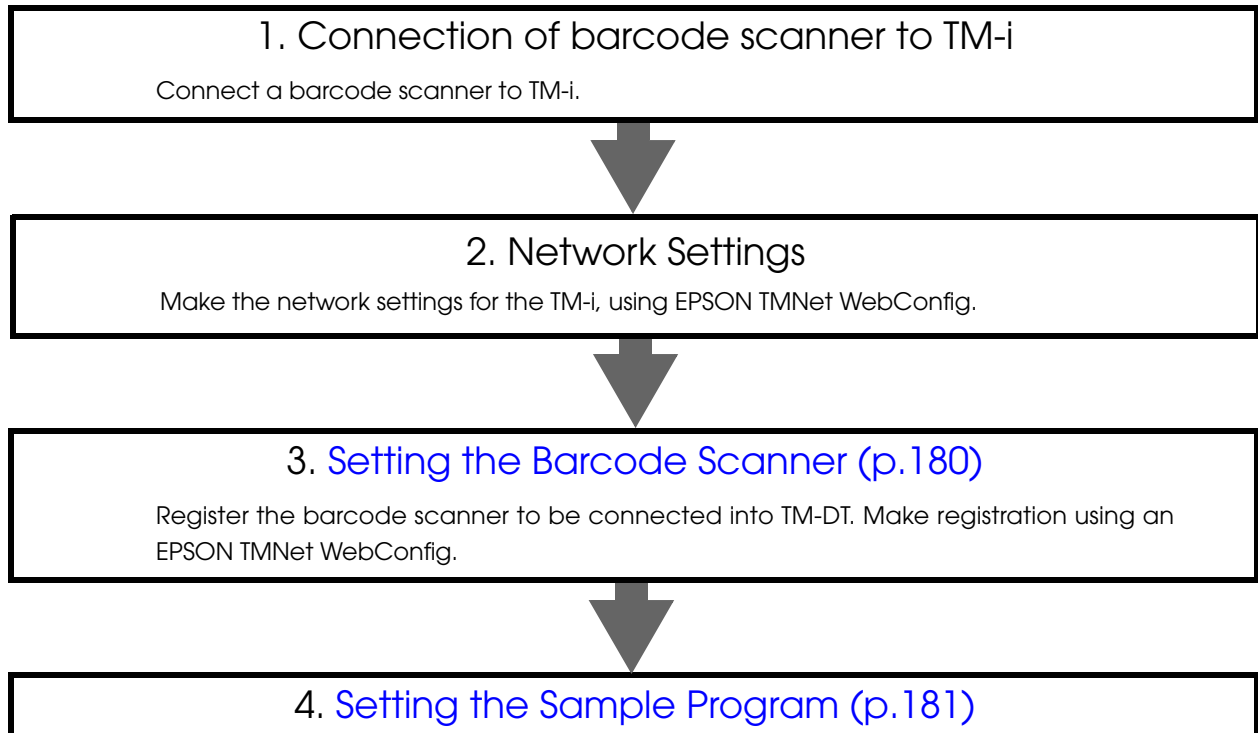
Register the barcode scanner to be connected into TM-DT. Make registration using a Web browser.



4. [Setting the Sample Program \(p.181\)](#)

Flow of environment construction for TM-i Series

The following illustrates the flow of constructing the environment for the sample program.



Setting the Barcode Scanner

Register the barcode scanner to be controlled by ePOS-Device XML into TM-Intelligent. Configure the settings using EPSON TMNet WebConfig.

EPSON TMNet WebConfig can be launched from your web browser.



For details on settings used, refer to the Technical Reference Guide for each TM-Intelligent.

Set the following.

Setting	Setting Value
Device ID	local_scanner
Device name	Select the relevant device from the list.
Control Script	Scanner_Generic.js

Setting the Sample Program

Configure the settings in the following procedure:

- 1** Expand ePOS-Device_Sample_XML_Vx.x.x.zip.
- 2** Start XCode and open the (ePOS-DeviceXMLSample.xcodeproj) project file.
- 3** When the project is opened, the project settings and source files will be displayed.
- 4** From the options displayed when (Set Active Executable) is selected, select an iPhone or iPad file.

Executing Sample Program



The iOS can be run with iOS Simulator.

Android

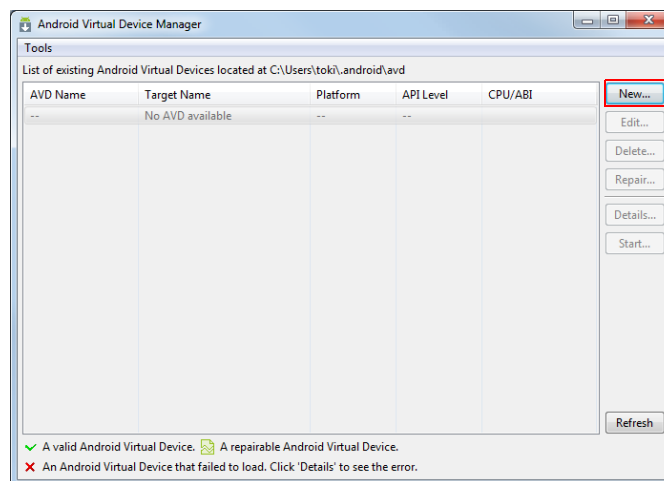
You can execute the sample program using the following methods.

- ❑ Executing with an emulator([p.182](#))
- ❑ Executing with an Android device([p.185](#))

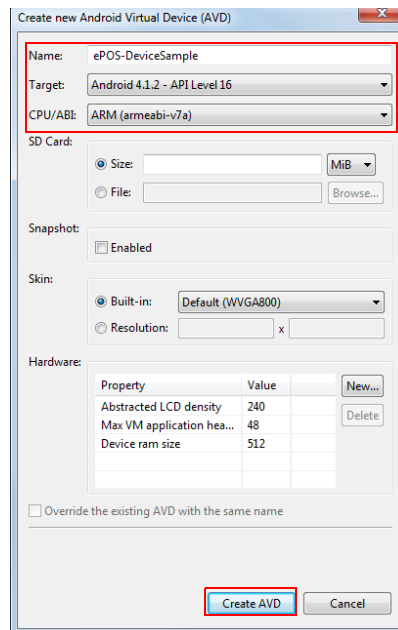
Executing with an emulator

Execute the program using the following procedure.

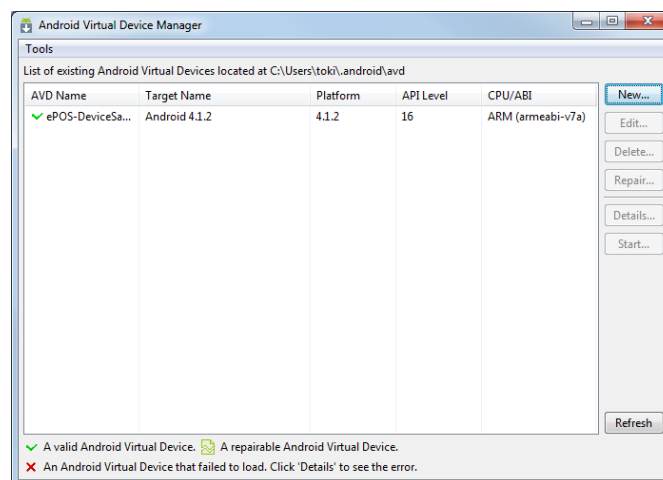
- 1 Start Android Virtual Device Manager. After it starts, click the (New...) button.
(Start)-(All Programs)-(Android SDK Tools)-(AVD Manager)



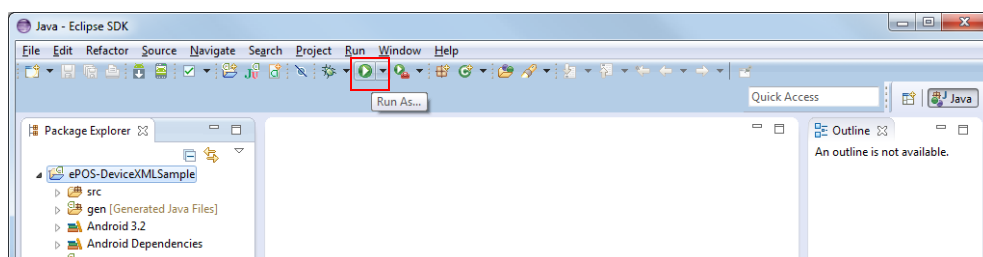
- 2** Select the name of the virtual device to create and the OS version of the virtual device. Input any value for (Name), and make selections for (Target) and (CPU/ABI). Then click the (Create AVD) button.



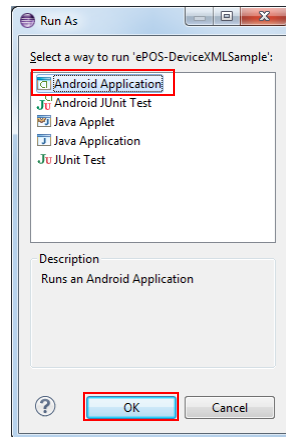
- 3** The created virtual device is displayed.



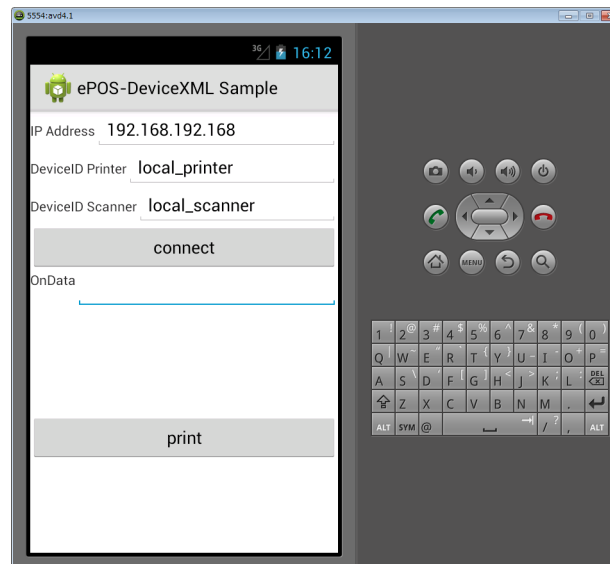
- 4** Click the button to run Eclipse.



- 5** For the initial startup, the window to confirm the execution type appears. Select (Android Application) and click the (OK) button.



- 6** The emulator starts and the application is executed.



Executing with an Android device

Execute the program using the following procedure.

- 1 Install the USB driver for the Android Debug Bridge (ADB) for the Android device you are using.

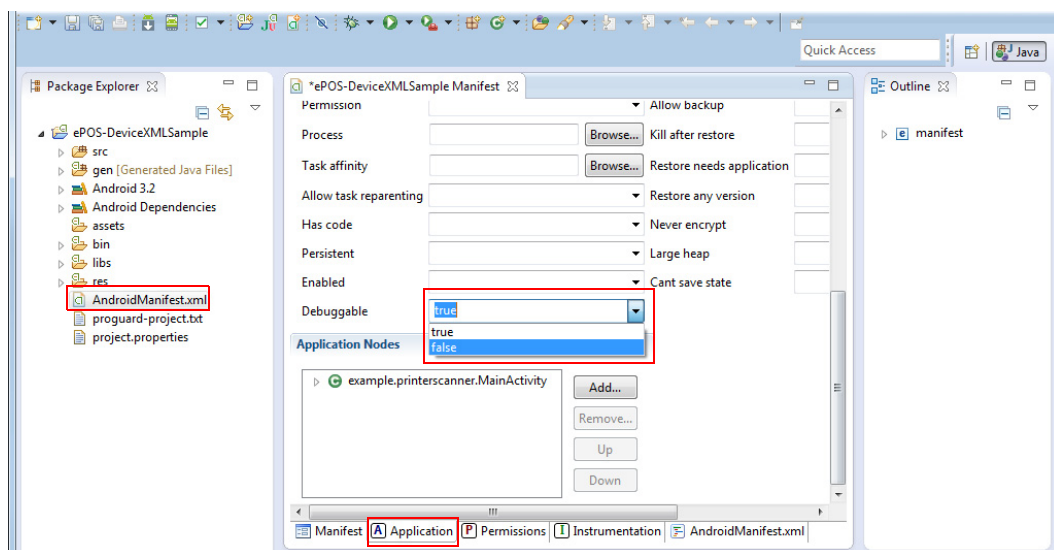
For more information about USB drivers for Android devices and their installation, see the site at the following URL.

<http://developer.android.com/tools/extras/oem-usb.html>

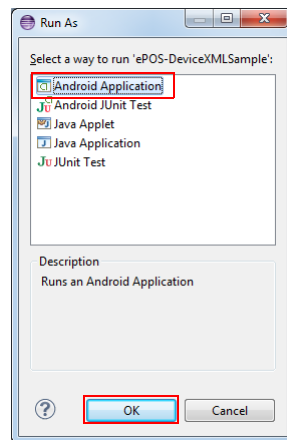


If you are unable to find an ADB USB driver for the device you are using at the site linked to above, search the website of the manufacturer of your device.

- 2 Make the following settings on your Android device:
 - Go to (Settings) - (Applications) , and enable (Unknown sources).
 - Select (Settings) - (Applications) - (Development) , and enable (USB debugging).
- 3 Connect the Android device to your computer via USB.
- 4 Enable the AndroidManifest.xml debugging file in the sample application. Double-click (AndroidManifest.xml). On the (Application) tab, set (Debuggable) to "true", and save the settings.



- 5** Click the Run button in Eclipse.
If it is being run for the first time, the following window will appear. Select (Android Application), and click the (OK) button.



- 6** The sample application is installed and run on the Android device.

How to Use the Sample Program

Use the sample program using the following procedure.

1 Perform application settings.

ePOS-DeviceXML Sample

IP Address 192.168.192.168

DeviceID Printer local_printer

DeviceID Scanner local_scanner

connect

OnData

print

Set as follows.

Setting Item	Description
IP Address	Input the TM-Intelligent's IP address.
DeviceID Printer	Input "local_printer".
DeviceID Scanner	Input "local_scanner".

2 Click or tap (connect).

3 Confirm that "Connect to server success" is displayed on the bottom of the screen.

IP Address 192.168.192.168

DeviceID Printer local_printer

DeviceID Scanner local_scanner

disconnect

OnData

print

local_scanner : OK
local_printer : OK
Connect to server success.

4 Scan a barcode using the barcode scanner.

- 5** If the barcode is properly scanned, the scan result is displayed in (OnData).

ePOS-DeviceXML Sample

IP Address 192.168.192.168

DeviceID Printer local_printer

DeviceID Scanner local_scanner

disconnect

OnData

1000000001044

1000000001037

1000000001020

1000000001013

1000000001051

- 6** Click or tap (print) to print the scan result.

OnData

1000000001044

1000000001037

1000000001020

1000000001013

1000000001051

print

- 7** If it is printed properly, "Print success" is displayed on the bottom of the screen.

OnData

1000000001044

1000000001037

1000000001020

1000000001013

1000000001051

print

Print success.

local_scanner : OK

local_printer : OK

Connect to server success.

Appendix

List of KeyCode

Key	KeyCode (Decimal)	Key	KeyCode (Decimal)
Backspace	8	Numbers on the keyboard	
Tab	9	4	52
Enter	13	5	53
Shift	16	6	54
Ctrl	17	7	55
Alt	18	8	56
Caps Lock	20	9	57
Esc	27	Alphabets on the keyboard	
Space	32	A	65
PgUp	33	B	66
PgDn	34	C	67
End	35	D	68
Home	36	E	69
!	37	F	70
#	38	G	71
i	39	H	72
\$	40	I	73
Insert	45	J	74
Delete	46	K	75
Numbers on the keyboard		L	76
0	48	M	77
1	49	N	78
2	50	O	79
3	51	P	80

Key	KeyCode (Decimal)	Key	KeyCode (Decimal)
Alphabets on the keyboard		Function key	
Q	81	F6	117
R	82	F7	118
S	83	F8	119
T	84	F9	120
U	85	F10	121
V	86	F11	122
W	87	F12	123
X	88	:	186
Y	89	;	187
Z	90	,	188
Numeric keypad		-	189
*	106	.	190
+	107	/	191
/	109	@	192
Function key		(	219
F1	112	\	220
F2	113	)	221
F3	114	^	222
F4	115	Caps Lock	240
F5	116		

Printer specifications

TM-H6000IV-DT/ TM-H6000IV

		Receipt		Slip	Endorsement
		80 mm	58 mm		
Resolution		180 x 180 dpi		-	
Print Width		512 dots		-	
Characters in a Line	Font A	ANK: 42 characters	ANK: 30 characters	ANK: 45 characters	ANK: 25 characters
	Font B	ANK: 56 characters	ANK: 40 characters	ANK: 60 characters	ANK: 33 characters
Character Size	Font A	ANK: 12 dots x 24 dots (W x H)		ANK: 5 dots x 9 dots (W x H)	
	Font B	ANK: 9 dots x 17 dots (W x H)		ANK: 7 dots x 9 dots (W x H)	
Character Baseline	Font A	At the 21st dot from the top of the character		-	
	Font B	At the 16th dot from the top of the character		-	
Default Line Feed Space		30 dots			
Color Specification		First color			
Page Mode Default Area		512 dots x 831 dots (W x H)	360 dots x 831 dots (W x H)	-	
Page Mode Maximum Area		512 dots x 1662 dots (W x H)	360 dots x 1662 dots (W x H)	-	
Raster Image <image>		Monochrome image, Gray scale image		Monochrome image	Not supported
Logo <logo>		Monochrome image, Gray scale image		Not supported	
Barcode <barcode>		UPC-A, UPC-E, EAN13, JAN13, EAN8, JAN8, CODE39, ITF, CODABAR, CODE93, CODE128, GS1-128, GS1 DataBar Omnidirectional, GS1 DataBar Truncated, GS1 DataBar Limited, GS1 DataBar Expanded			-

	Receipt		Slip	Endorsement
	80 mm	58 mm		
2D-Code <symbol>	PDF417, QR Code, MaxiCode, GS1 DataBar Stacked, GS1 DataBar Stacked Omnidirectional, GS1 DataBar Expanded Stacked, Composite Symbology		Not supported	
Ruled Line <hline>, <vline-begin>, <vline-end>	Not supported			
Page Mode	Line <line>	Not supported		
	Rectangle <rectangle>			
Paper Cut <cut>	Cut, Feed cut		Not supported	
Drawer Kick-Out <pulse>	Supported			
Buzzer <sound>	Not supported			
Paper Layout Settings <layout>	Not supported			
Command <command>	Supported			
MICR	Supported			

TM-L90

		Receipt	Die-cut Label
Resolution		203 dpi x 203 dpi (W x H)	
Print Width		256 dots (38 mm) to 576 dots (80 mm)	224 dots (38 mm) to 560 dots (80 mm)
Characters in a Line	Font A	ANK: 48 characters,	ANK: 46 characters
	Font B	ANK: 57 characters	ANK: 56 characters
	Font C	ANK: 72 characters	ANK: 70 characters
Character Size	Font A	ANK: 12 dots x 24 dots (W x H)	
	Font B	ANK: 10 dots x 24 dots (W x H)	
	Font C	ANK: 8 dots x 16 dots (W x H)	
Character Baseline	Font A	At the 21st dot from the top of the character	
	Font B	At the 21st dot from the top of the character	
	Font C	At the 15st dot from the top of the character	
Default Line Feed Space		30 dots	
Color Specification		First color Second color, Second color (when two-color printing is set)	
Page Mode Default Area		576 dots x 738 dots (W x H)	560 dots x 738 dots (W x H)
	when two-color printing is set	576 dots x 369 dots (W x H)	560 dots x 369 dots (W x H)
Page Mode Maximum Area		576 dots x 1476 dots (W x H)	560 dots x 1476 dots (W x H)
	when two-color printing is set	576 dots x 738 dots (W x H)	560 dots x 738 dots (W x H)
Raster Image <image>		Monochrome image, Two color image	
Logo <logo>		Monochrome image, Two color image (To perform two-color printing, change the settings of the printer using the memory switch setting utility.)	
Barcode <barcode>		UPC-A, UPC-E, EAN13, JAN13, EAN8, JAN8, CODE39, ITF, CODABAR, CODE93, CODE128	
2D-Code <symbol>		PDF417, QR Code, MaxiCode	
Ruled Line <hline>, <vline-begin>, <vline-end>		Not supported	

		Receipt	Die-cut Label
Page Mode	Line <line>	Not supported	
	Rectangle <rectangle>		
Paper Cut <cut>		Cut, Feed cut	
Drawer Kick-Out <pulse>		Supported	
Buzzer <sound>		Not supported	
Paper Layout Settings <layout>		Not supported (With automatic setup mode)	
Command <command>		Supported	

TM-P20

		58 mm
Resolution		203 dpi x 203 dpi (W x H)
Print Width		384 dots
Characters in a Line	Font A	ANK: 32 characters
	Font B	ANK: 42 characters
	Font C	ANK: 42 characters
	Font D	ANK: 38 characters
	Font E	ANK: 48 characters
Character Size	Font A	ANK: 12 dots x 24 dots (W x H)
	Font B	ANK: 9 dots x 24 dots (W x H)
	Font C	ANK: 9 dots x 17 dots (W x H)
	Font D	ANK: 10 dots x 24 dots (W x H)
	Font E	ANK: 8 dots x 16 dots (W x H)
Character Baseline	Font A	At the 21st dot from the top of the character
	Font B	At the 16th dot from the top of the character
	Font C	At the 16th dot from the top of the character
	Font D	At the 21st dot from the top of the character
	Font E	At the 15th dot from the top of the character
Default Line Feed Space		30 dots
Color Specification		First color
Page Mode Default Area		384 dots x 2400 dots (W x H)
Page Mode Maximum Area		384 dots x 2400 dots (W x H)
Raster Image <image>		Monochrome image
Logo <logo>		Monochrome image
Barcode <barcode>		UPC-A, UPC-E, EAN13, JAN13, EAN8, JAN8, CODE39, ITF, CODABAR, CODE93, CODE128, GS1-128, GS1 DataBar Omnidirectional, GS1 DataBar Truncated, GS1 DataBar Limited, GS1 DataBar Expanded
2D-Code <symbol>		PDF417, QR Code, MaxiCode, GS1 DataBar Stacked, GS1 DataBar Stacked Omnidirectional, GS1 DataBar Expanded Stacked, Composite Symbology
Ruled Line <hline>, <vline-begin>, <vline-end>		Not supported

		58 mm
Page Mode	Line <line>	Supported (Only solid line)
	Rectangle <rectangle>	
Paper Cut <cut>		Feeds paper to cutting position
Drawer Kick-Out <pulse>		Not supported
Buzzer <sound>		Supported
Paper Layout Settings <layout>		Supported
Command <command>		Supported

TM-P60II/ TM-P60II with Peeler

		Receipt	Die-cut label
Resolution		203 dpi x 203 dpi (W x H)	
Print Width		432 dots	160 dots ~ 400 dots
Characters in a Line	Font A	ANK: 36 characters	ANK: 33 characters
	Font B	ANK: 43 characters	ANK: 40 characters
	Font C	ANK: 54 characters	ANK: 50 characters
Character Size	Font A	ANK: 12 dots x 24 dots (W x H)	
	Font B	ANK: 10 dots x 24 dots (W x H)	
	Font C	ANK: 8 dots x 16 dots (W x H)	
Character Baseline	Font A	At the 21st dot from the top of the character	
	Font B	At the 21st dot from the top of the character	
	Font C	At the 15th dot from the top of the character	
Default Line Feed Space		30 dots	
Color Specification		First color	
Page Mode Default Area		432 dots x 1624 dots (W x H)	400 dots x 1624 dots (W x H)
Page Mode Maximum Area		432 dots x 1624 dots (W x H)	400 dots x 1624 dots (W x H)
Raster Image <image>		Monochrome image	
Logo <logo>		Monochrome image	
Barcode <barcode>		UPC-A, UPC-E, EAN13, JAN13, EAN8, JAN8, CODE39, ITF, CODABAR, CODE93, CODE128, GS1-128, GS1 DataBar Omnidirectional, GS1 DataBar Truncated, GS1 DataBar Limited, GS1 DataBar Expanded	
2D-Code <symbol>		PDF417, QR Code, MaxiCode, GS1 DataBar Stacked, GS1 DataBar Stacked Omnidirectional, GS1 DataBar Expanded Stacked, Aztec Code, DataMatrix (Composit Symbology : Not supported)	
Ruled Line <hline>, <vline-begin>, <vline-end>		Not supported	
Page Mode	Line <line>	Supported (Only solid line)	
	Rectangle <rectangle>		
Paper Cut <cut>		Feed cut (Feeds paper to cutting position)	
Drawer Kick-Out <pulse>		Not supported	
Buzzer <sound>		Support (Pattern 1 ~ Pattern 10, Stop)	

	Receipt	Die-cut label
Paper Layout Settings <layout>	Supported	
Command <command>	Supported	

TM-P80

		80 mm
Resolution		203 dpi x 203 dpi (W x H)
Print Width		576 dots
Characters in a Line	Font A	ANK: 48 characters
	Font B	ANK: 64 characters
Character Size	Font A	ANK: 12 dots x 24 dots (W x H)
	Font B	ANK: 9 dots x 17 dots (W x H)
Character Baseline	Font A	At the 21st dot from the top of the character
	Font B	At the 16th dot from the top of the character
Default Line Feed Space		30 dots
Color Specification		First color
Page Mode Default Area		576 dots x 1662 dots (W x H)
Page Mode Maximum Area		576 dots x 1662 dots (W x H)
Raster Image <image>		Monochrome image
Logo <logo>		Monochrome image
Barcode <barcode>		UPC-A, UPC-E, EAN13, JAN13, EAN8, JAN8, CODE39, ITF, CODABAR, CODE93, CODE128, GS1-128, GS1 DataBar Omnidirectional, GS1 DataBar Truncated, GS1 DataBar Limited, GS1 Databar Expanded
2D-Code <symbol>		PDF417, QR Code, MaxiCode, Data Matrix, Aztec Code, GS1 DataBar Stacked, GS1 DataBar Stacked Omnidirectional, GS1 DataBar Expanded Stacked, Composite Symbology
Ruled Line <hline>, <vline-begin>, <vline-end>		Not supported
Page Mode	Line <line>	Not supported
	Rectangle <rectangle>	
Paper Cut <cut>		Feed
Drawer Kick-Out <pulse>		Not supported
Buzzer <sound>		Supported
Paper Layout Settings <layout>		Supported
Command <command>		Supported

TM-T20

		58 mm	80 mm
Resolution		203 dpi x 203 dpi (W x H)	
Print Width		420 dots	576 dots
Characters in a Line	Font A	ANK: 35 characters,	ANK: 48 characters
	Font B	ANK: 46 characters	ANK: 64 characters
Character Size	Font A	ANK: 12 dots x 24 dots (W x H)	
	Font B	ANK: 9 dots x 17 dots (W x H)	
Character Baseline	Font A	At the 21st dot from the top of the character	
	Font B	At the 16th dot from the top of the character	
Default Line Feed Space		30 dots	
Color Specification		First color	
Page Mode Default Area		420 dots x 831 dots (W x H)	576 dots x 831 dots (W x H)
Page Mode Maximum Area		420 dots x 1662 dots (W x H)	576 dots x 1662 dots (W x H)
Raster Image <image>		Monochrome image	
Logo <logo>		Monochrome image	
Barcode <barcode>		UPC-A, UPC-E, EAN13, JAN13, EAN8, JAN8, CODE39, ITF, CODABAR, CODE93, CODE128, GS1-128, GS1 DataBar Omnidirectional, GS1 DataBar Truncated, GS1 DataBar Limited, GS1 DataBar Expanded	
2D-Code <symbol>		PDF417, QR Code, MaxiCode, GS1 DataBar Stacked, GS1 DataBar Stacked Omnidirectional, GS1 DataBar Expanded Stacked (Composit Symbology : Not supported)	
Ruled Line <hline>, <vline-begin>, <vline-end>		Not supported	
Page Mode	Line <line>	Not supported	
	Rectangle <rectangle>		
Paper Cut <cut>		Cut, Feed cut	
Drawer Kick-Out <pulse>		Supported	
Buzzer <sound>		Supported (Pattern A ~ Pattern E, Error, No paper, Stop)	
Paper Layout Settings <layout>		Not supported	
Command <command>		Supported	

TM-T20II-i/ TM-T20II

	58 mm	80 mm
Resolution	203 dpi x 203 dpi (W x H)	
Print Width (42 Column Mode)	420 dots (378 dots)	576 dots (546 dots)
Characters in a Line Font A (42 Column Mode)	ANK: 35 characters (ANK: 42 characters)	ANK: 48 characters (ANK: 42 characters)
Font B (42 Column Mode)	ANK: 46 characters (ANK: 31 characters)	ANK: 64 characters (ANK: 60 characters)
Character Size Font A (42 Column Mode)	ANK: 12 dots x 24 dots (W x H)	
	(ANK: 9 dots x 17 dots)	(ANK: 13 dots x 24 dots)
Font B (42 Column Mode)	ANK: 9 dots x 17 dots (W x H)	
	(ANK: 12 dots x 24 dots)	(ANK: 9 dots x 17 dots)
Character Baseline Font A	At the 21st dot from the top of the character	
Font B	At the 16th dot from the top of the character	
Default Line Feed Space	30 dots	
Color Specification	First color	
Page Mode Default Area	420 dots x 831 dots (W x H)	576 dots x 831 dots (W x H)
Page Mode Maximum Area	420 dots x 1662 dots (W x H)	576 dots x 1662 dots (W x H)
Raster Image <image>	Monochrome image	
Logo <logo>	Monochrome image	
Barcode <barcode>	UPC-A, UPC-E, EAN13, JAN13, EAN8, JAN8, CODE39, ITF, CODABAR, CODE93, CODE128, GS1-128, GS1 DataBar Omnidirectional, GS1 DataBar Truncated, GS1 DataBar Limited, GS1 Databar Expanded	
2D-Code <symbol>	PDF417, QR Code, MaxiCode, GS1 DataBar Stacked, GS1 DataBar Stacked Omnidirectional, GS1 DataBar Expanded Stacked, Composite Symbology	
Ruled Line <hline>, <vline-begin>, <vline-end>	Not supported	
Page Mode Line <line>	Not supported	
Rectangle <rectangle>		
Paper Cut <cut>	Cut, Feed cut	
Drawer Kick-Out <pulse>	Supported	

	58 mm	80 mm
Buzzer <sound>	Supported	
Paper Layout Settings <layout>	Not supported	
Command <command>	Supported	

TM-T70-i/ TM-T70

		58 mm	80 mm
Resolution		203 dpi x 203 dpi (W x H)	
Print Width		420 dots	576 dots
Characters in a Line	Font A	ANK: 34 characters	ANK: 48 characters
	Font B	ANK: 46 characters	ANK: 64 characters
Character Size	Font A	ANK: 12 dots x 24 dots (W x H)	
	Font B	ANK: 8 dots x 16 dots (W x H)	
Character Baseline	Font A	At the 21st dot from the top of the character	
	Font B	At the 15th dot from the top of the character	
Default Line Feed Space		30 dots	
Color Specification		First color	
Page Mode Default Area		576 dots x 1662 dots (W x H)	
Page Mode Maximum Area		576 dots x 1662 dots (W x H)	
Raster Image <image>		Monochrome image	
Logo <logo>		Monochrome image	
Barcode <barcode>		UPC-A, UPC-E, EAN13, JAN13, EAN8, JAN8, CODE39, ITF, CODABAR, CODE93, CODE128	
2D-Code <symbol>		PDF417, QR Code	
Ruled Line <hline>, <vline-begin>, <vline-end>		Not supported	
Page Mode	Line <line>	Not supported	
	Rectangle <rectangle>		
Paper Cut <cut>		Cut, Feed cut	
Drawer Kick-Out <pulse>		Supported	
Buzzer <sound>		Not supported	
Paper Layout Settings <layout>		Not supported	
Command <command>		Supported	

TM-T70II-DT/ TM-T70II

		80 mm
Resolution		203 dpi x 203 dpi (W x H)
Print Width		576 dots
Characters in a Line	Font A	ANK: 48 characters
	Font B	ANK: 72 characters
Character Size	Font A	ANK: 12 dots x 24 dots (W x H)
	Font B	ANK: 9 dots x 17 dots (W x H)
Character Baseline	Font A	At the 21st dot from the top of the character
	Font B	At the 15th dot from the top of the character
Default Line Feed Space		30 dots
Color Specification		First color
Page Mode Default Area		576 dots x 1662 dots (W x H)
Page Mode Maximum Area		576 dots x 1662 dots (W x H)
Raster Image <image>		Monochrome image
Logo <logo>		Monochrome image
Barcode <barcode>		UPC-A, UPC-E, EAN13, JAN13, EAN8, JAN8, CODE39, ITF, CODABAR, CODE93, CODE128, GS1-128, GS1 DataBar Omnidirectional, GS1 DataBar Truncated, GS1 DataBar Limited, GS1 Databar Expanded
2D-Code <symbol>		PDF417, QR Code, MaxiCode, GS1 DataBar Stacked, GS1 DataBar Stacked Omnidirectional, GS1 DataBar Expanded Stacked
Ruled Line <hline>, <vline-begin>, <vline-end>		Not supported
Page Mode	Line <line>	Not supported
	Rectangle <rectangle>	
Paper Cut <cut>		Cut, Feed cut
Drawer Kick-Out <pulse>		Supported
Buzzer <sound>		Supported
Paper Layout Settings <layout>		Not supported
Command <command>		Supported

TM-T82II-i/ TM-T82II

	80 mm
Resolution	203 dpi x 203 dpi (W x H)
Print Width (42 Column Mode)	576 dots (546 dots)
Characters in a Line Font A (42 Column Mode)	ANK: 48 characters (ANK: 42 characters)
Font B (42 Column Mode)	ANK: 64 characters (ANK: 60 characters)
Character Size Font A (42 Column Mode)	ANK: 12 dots x 24 dots (W x H) (ANK: 13 dots x 24 dots)
Font B (42 Column Mode)	ANK: 9 dots x 17 dots (W x H) (ANK: 9 dots x 17 dots)
Character Baseline Font A	At the 21st dot from the top of the character
Font B	At the 16th dot from the top of the character
Default Line Feed Space	30 dots
Color Specification	First color
Page Mode Default Area	576 dots x 831 dots (W x H)
Page Mode Maximum Area	576 dots x 1662 dots (W x H)
Raster Image <image>	Monochrome image
Logo <logo>	Monochrome image
Barcode <barcode>	UPC-A, UPC-E, EAN13, JAN13, EAN8, JAN8, CODE39, ITF, CODABAR, CODE93, CODE128, GS1-128, GS1 DataBar Omnidirectional, GS1 DataBar Truncated, GS1 DataBar Limited, GS1 Databar Expanded
2D-Code <symbol>	PDF417, QR Code, MaxiCode, GS1 DataBar Stacked, GS1 DataBar Stacked Omnidirectional, GS1 DataBar Expanded Stacked, Composite Symbology
Ruled Line <hline>, <vline-begin>, <vline-end>	Not supported
Page Mode Line <line>	Not supported
Rectangle <rectangle>	

	80 mm
Paper Cut <cut>	Cut, Feed cut
Drawer Kick-Out <pulse>	Supported
Buzzer <sound>	Supported
Paper Layout Settings <layout>	Not supported
Command <command>	Supported

TM-T83II-i/ TM-T83II

		80 mm
Resolution		180 dpi x 180 dpi (W x H)
Print Width		512 dots
Characters in a Line	Font A	ANK: 42 characters Kanji: 21 characters
	Font B	ANK: 56 characters Kanji: 32 characters
Character Size	Font A	ANK: 12 dots x 24 dots (W x H) Kanji: 24 dots x 24 dots (W x H)
	Font B	ANK: 9 dots x 17 dots (W x H) Kanji: 16 dots x 16 dots (W x H)
Character Baseline	Font A	ANK: At the 21st dot from the top of the character Kanji: At the 21st dot from the top of the character
	Font B	ANK: At the 16th dot from the top of the character Kanji: At the 15th dot from the top of the character
Default Line Feed Space		30 dots
Color Specification		First color
Page Mode Default Area		512 dots x 1662 dots (W x H)
Page Mode Maximum Area		512 dots x 1662 dots (W x H)
Raster Image <image>		Monochrome image
Logo <logo>		Monochrome image
Barcode <barcode>		UPC-A, UPC-E, EAN13, JAN13, EAN8, JAN8, CODE39, ITF, CODABAR, CODE93, CODE128, GS1-128, GS1 DataBar Omnidirectional, GS1 DataBar Truncated, GS1 DataBar Limited, GS1 DataBar Expanded
2D-Code <symbol>		PDF417, QR Code, MaxiCode, GS1 DataBar Stacked, GS1 DataBar Stacked Omnidirectional, GS1 DataBar Expanded Stacked
Ruled Line <hline>, <vline-begin>, <vline-end>		Not supported
Page Mode	Line <line>	Not supported
	Rectangle <rectangle>	
Paper Cut <cut>		Cut, Feed cut
Drawer Kick-Out <pulse>		Supported

	80 mm
Buzzer <sound>	Supported
Paper Layout Settings <layout>	Not supported
Command <command>	Supported

TM-T88IV

		80 mm	58 mm
Resolution		180 dpi x 180 dpi (W x H)	
Print Width		512 dots	360 dots
Characters in a Line	Font A	ANK: 42 characters	ANK: 30 characters
	Font B	ANK: 56 characters	ANK: 40 characters
Character Size	Font A	ANK: 12 dots x 24 dots (W x H)	
	Font B	ANK: 9 dots x 17 dots (W x H)	
Character Baseline	Font A	At the 21st dot from the top of the character	
	Font B	At the 16th dot from the top of the character	
Default Line Feed Space		30 dots	
Color Specification		First color First color, Second color (when two-color printing is set)	
Page Mode Default Area		512 dots x 831 dots (W x H)	360 dots x 831 dots (W x H)
	when two-color printing is set	512 dots x 415 dots (W x H)	360 dots x 415 dots (W x H)
Page Mode Maximum Area		512 dots x 1662 dots (W x H)	360 dots x 1662 dots (W x H)
	when two-color printing is set	512 dots x 831 dots (W x H)	360 dots x 831 dots (W x H)
Raster Image <image>		Monochrome image, two-color image	
Logo <logo>		Monochrome image, two-color image (To perform two-color printing, change the settings of the printer using the memory switch setting utility.)	
Bar Code		UPC-A, UPC-E, EAN13, JAN13, EAN8, JAN8, CODE39, ITF, CODABAR, CODE93, CODE128, GS1-128	
2D-Code <symbol>		PDF417, QR Code	
Ruled Line <hline>, <vline-begin>, <vline-end>		Not supported	
Page Mode	Line <line>	Not supported	
	Rectangle <rectangle>		
Paper Cut <cut>		Cut, Feed cut	
Drawer Kick-Out <pulse>		Supported	
Buzzer <sound>		Not supported	
Paper Layout Settings <layout>		Not supported	

	80 mm	58 mm
Command <command>	Supported	

TM-T88V-DT/ TM-T88V-i/ TM-T88V

* Only for South Asia model

		58 mm	80 mm
Resolution		180 dpi x 180 dpi (W x H)	
Print Width		360 dots	512 dots
Characters in a Line	Font A	ANK: 30 characters	ANK: 42 characters
	Font B	ANK: 40 characters	ANK: 56 characters
	Special font A*	30 characters	42 characters
	Special font B*	40 characters	56 characters
Character Size	Font A	ANK: 12 dots x 24 dots (W x H)	
	Font B	ANK: 9 dots x 17 dots (W x H)	
	Special font A*	12 dots x 24 dots (W x H)	
	Special font B*	9 dots x 24 dots (W x H)	
Character Baseline	Font A	At the 21st dot from the top of the character	
	Font B	At the 16th dot from the top of the character	
	Special font A*	At the 20th dot from the top of the character	
	Special font B*	At the 20th dot from the top of the character	
Default Line Feed Space		30 dots	
Color Specification		First color	
Page Mode Default Area		360 dots x 831 dots (W x H)	512 dots x 831 dots (W x H)
Page Mode Maximum Area		360 dots x 1662 dots (W x H)	512 dots x 1662 dots (W x H)
Raster Image <image>		Monochrome image, Gray scale image	
Logo <logo>		Monochrome image, Gray scale image	
Barcode <barcode>		UPC-A, UPC-E, EAN13, JAN13, EAN8, JAN8, CODE39, ITF, CODABAR, CODE93, CODE128, GS1-128, GS1 DataBar Omnidirectional, GS1 DataBar Truncated, GS1 DataBar Limited, GS1 Databar Expanded	
2D-Code <symbol>		PDF417, QR Code, MaxiCode, GS1 DataBar Stacked, GS1 DataBar Stacked Omnidirectional, GS1 DataBar Expanded Stacked (Composite Symbology not supported)	

* Only for South Asia model

	58 mm	80 mm
Ruled Line <hline>, <vline-begin>, <vline-end>	Not supported	
Page Mode	Line <line>	Not supported
	Rectangle <rectangle>	
Paper Cut <cut>	Cut, Feed cut	
Drawer Kick-Out <pulse>	Supported	
Buzzer <sound>	Supported	
Paper Layout Settings <layout>	Not supported	
Command <command>	Supported	

TM-T90

		58 mm	60 mm	80 mm
Resolution		180 dpi x 180 dpi (W x H)		
Print Width		360 dots	384 dots	512 dots
Characters in a Line	Font A	ANK: 30 characters,	ANK: 32 characters	ANK: 42 characters
	Font B	ANK: 40 characters	ANK: 42 characters	ANK: 56 characters
	Font C	ANK: 52 characters	ANK: 54 characters	ANK: 72 characters
Character Size	Font A	ANK: 12 dots x 24 dots (W x H)		
	Font B	ANK: 9 dots x 17 dots (W x H)		
	Font C	ANK: 8 dots x 16 dots (W x H)		
Character Baseline	Font A	At the 21st dot from the top of the character		
	Font B	At the 16th dot from the top of the character		
	Font C	At the 15th dot from the top of the character		
Default Line Feed Space		30 dots		
Color Specification		First color Second color, Second color (when two-color printing is set)		
Page Mode Default Area		360 dots x 831 dots (W x H)	384 dots x 831 dots (W x H)	512 dots x 831 dots (W x H)
	when two-color printing is set	360 dots x 415 dots (W x H)	384 dots x 415 dots (W x H)	512 dots x 415 dots (W x H)
Page Mode Maximum Area		360 dots x 1662 dots (W x H)	384 dots x 1662 dots (W x H)	512 dots x 1662 dots (W x H)
	when two-color printing is set	360 dots x 831 dots (W x H)	384 dots x 831 dots (W x H)	512 dots x 831 dots (W x H)
Raster Image <image>		Monochrome image, Two color image		
Logo <logo>		Monochrome image, Two color image (To perform two-color printing, change the settings of the printer using the memory switch setting utility.)		
Barcode <barcode>		UPC-A, UPC-E, EAN13, JAN13, EAN8, JAN8, CODE39, ITF, CODABAR, CODE93, CODE128		
2D-Code <symbol>		PDF417		

	58 mm	60 mm	80 mm
Ruled Line <hline>, <vline-begin>, <vline-end>	Not supported		
Page Mode	Line <line>	Not supported	
	Rectangle <rectangle>		
Paper Cut <cut>	Cut, Feed cut		
Drawer Kick-Out <pulse>	Supported		
Buzzer <sound>	Not supported		
Paper Layout Settings <layout>	Not supported		
Command <command>	Supported		

TM-U220-i/ TM-U220

		76 mm	70 mm	58 mm
Resolution		80 dpi x 72 dpi (W x H)		
Print Width		200 dots	180 dots	150 dots
Characters in a Line	Font A	ANK: 33 characters,	ANK: 30 characters	ANK: 25 characters
	Font B	ANK: 40 characters	ANK: 36 characters	ANK: 30 characters
Character Size	Font A	ANK: 4.5 dots x 9 dots (W x H)		
	Font B	ANK: 3.5 dots x 9 dots (W x H)		
Character Baseline	Font A	Bottom of the characters		
	Font B	Bottom of the characters		
Default Line Feed Space		12 dots		
Color Specification		First color First color, Second color (When using a two-color ribbon cassette)		
Page Mode Default Area		Not supported		
Page Mode Maximum Area		Not supported		
Raster Image <image>		Monochrome image		
Logo <logo>		Not supported		
Barcode <barcode>		Not supported		
2D-Code <symbol>		Not supported		
Ruled Line <hline>, <vline-begin>, <vline-end>		Not supported		
Page Mode	Line <line>	Not supported		
	Rectangle <rectangle>			
Paper Cut <cut>		Cut, Feed cut		
Drawer Kick-Out <pulse>		Supported		
Buzzer <sound>		Not supported		
Paper Layout Settings <layout>		Not supported		
Command <command>		Supported		

